JENNY SCIENCE präzis bewegen, auf engstem Raum

XENAX® Powerlink PLCopen Library

Version 1.2

June 2015

ETHERNET POWERLINK

XENAX® Ethernet Servo Controller with
POWERLINK® fieldbus module

Functional safety, TÜV certified

Force processes for “Force Control“

General

This manual describes the integration of the XENAX® Xvi75V8 Servo Controller into a B&R PLC with the Automation Studio. Therefore the Jenny Science PLCopen library “JsMcLib” based on the CANopen Motion Profile DS402 will be used.

This documantation includes examples with configurations, program integration and testing.

Contents

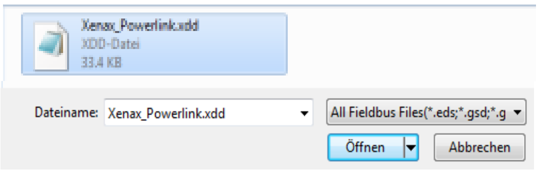
1 Development environment

1.1 Controller, Tools, Libraries

Automation Studio
Software for the configuration of the B&R PLC with project handling.

Software for configuration and programming of the B&R PLC, including project management. This manual is based on Automation Studio V4

Electronic data sheet
XDD-file (XML Device Description) is a standard XML-format according to ISO 15745-4. This file includes communication parameter and objects of the XENAX® Servo Controller. These are necessary for the integration into Automation Studo V4.

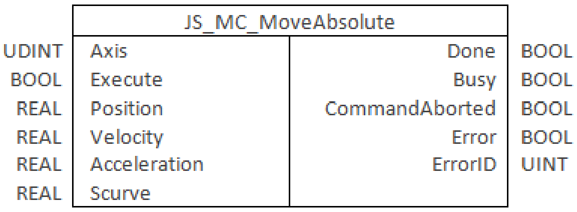


Xenax_Powerlink.xdd
XDD-Datei
33.4 KB

Dateiname: Xenax_Powerlink.xdd

All Fieldbus Files(*.eds;*.gsd;*.g

JsMcLib
(Jenny Science Motion Control Library)
This PLCopen library is based on the motion profile DS 402 including the corresponding "state-machine". Using the Jenny Science library will simplify the "motion"-programming.



JS_MC_MoveAbsolute

Type	Input	Output	Type
UDINT	Axis	Done	BOOL
BOOL	Execute	Busy	BOOL
REAL	Position	CommandAborted	BOOL
REAL	Velocity	Error	BOOL
REAL	Acceleration	ErrorID	UINT
REAL	Scurve		

B&R PLC with Powerlink
For example the X20-System, controls the cyclic data communication and defines the clock pulse for the "cyclic synchronous position mode" which is used for axis interpolation.

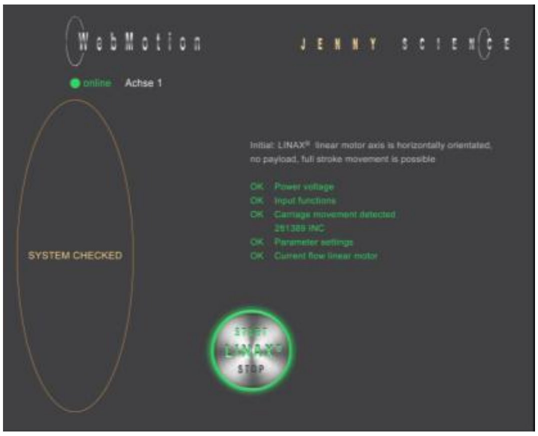
1.2 WebMotion®

WebMotion®

This is the graphical user interface from Jenny Science. It is stored in the embedded Web server of the XENAX® Servo Controller as a Java applet. WebMotion ® is launched with a web browser by entering the TCP/IP address of XENAX® Servo Controller.

LINAX® / ELAX® linear motor axes are automatically recognized. The corresponding controller parameters are saved and loaded automatically. With the Quick Start button, the linear motor axis can operate immediately. No user manual is needed.

.



Initial: LINAX® linear motor axis is horizontally orientated, no payload, full stroke movement is possible

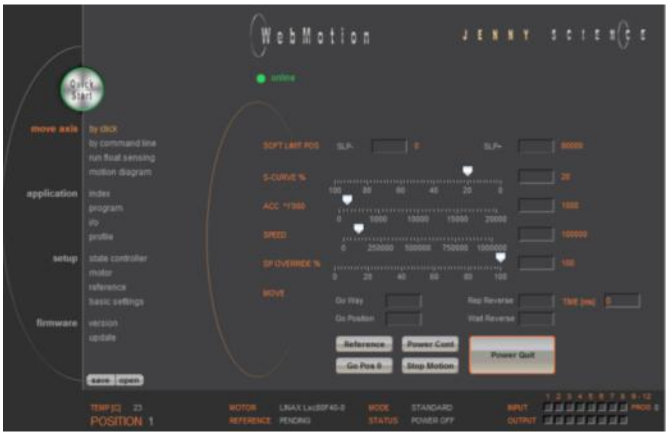
OK Power voltage
OK Input functions
OK Carriage movement detected
281389 INC
OK Parameter settings
OK Current flow linear motor

The parametrization of the XENAX® Servo Controller is made over an Ethernet TCP/IP connection.

You can find the pre-set IP-address on the backside of your XENAX® Servo Controller.

Web address:
http://192.168.2.1xx/XENAX.html
To change the IP/ Web address you can use the Lantronix DeviceInstaller.

For additional information of the TCP/IP connection you could use the "Xvi75_Manual_E.pdf"or the video Tutorial on YouTube.
http://www.jennyscience.de/video-tutorials/.

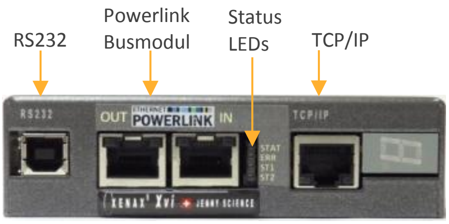


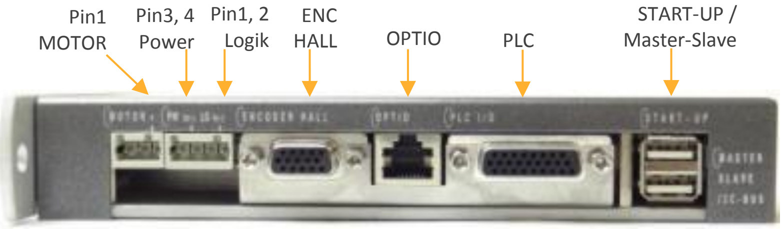
1.3 XENAX® Servo Controller

XENAX® Ethernet Servo Controller
With additional POWERLINK fieldbus module.

To use the JsMcLib v2.10.1 or higher the XENAX® Servo Controller firmware v3.64D or higher and the POWERLINK fieldbus module firmware v2.0 or higher are required.

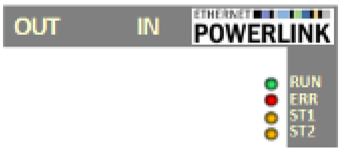
1.3.1 Socket XENAX®





1.3.2 State LEDs on the Powerlink fieldbus module.

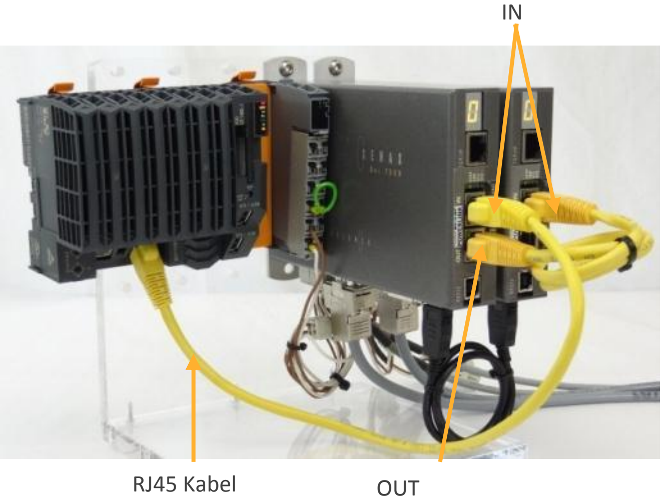
These are the states of the LED on the Powerlink fieldbus module.



LED state	**RUN** (STAT)	**ERR**	**ST1** Status 1	**ST2** Status 2
<OFF>	In init process or no power	Bus module no error		Bus module l ready
<ON>	Operational state	State bus off	No application in the flash	
<Blink>	Pre-Operational state			Protocol download in progress

1.4 Powerlink connection from B&R PLC to XENAX®

Typically the Powerlink fieldbus is controlled with a linear structure from device to device. Shielded RJ45 cables go from the IN to the OUT to connect the devices with each other.



1.5 Industrial Hub for more than 4 XENAX®

A maximum of 4 XENAX® Servo Controllers can be connected in a row.
If there are more than 4 Servo Controller used, an industrial hub from B&R will be necessary.
Each contact can be used for 4 Servo Controller.

For example: With the HUB Nr. 0AC808.9-1 from B&R.
A maximum of 7*4 = 28 controllers could be connected.

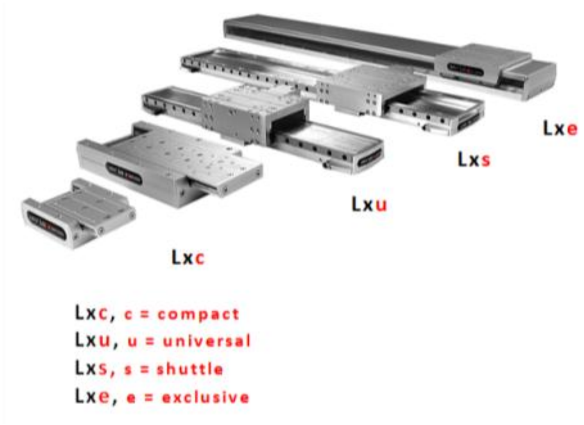
1.6 Firmware Update

The firmware can easily be updated with the according tutorial video “Ethernet TCP/IP Connection to XENAX” under www.jennyscience.de/en/ -> *Video Tutorials.*

1.7 LINAX®, ELAX® Linear Motor Axis

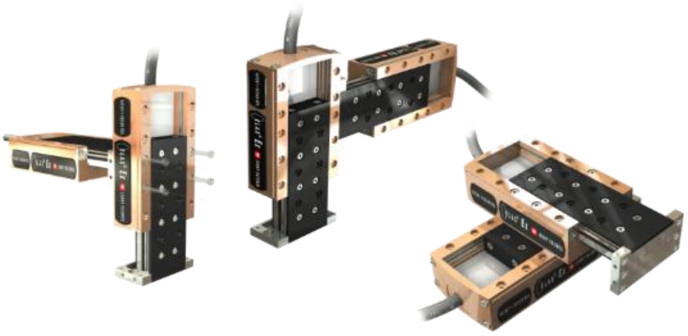
LINAX® Linear Motor-Axis

Are available in different lengths and types. The LINAX® linear motor axes are highly modular and can be combined flexibly amongst each other. The XENAX® Servo Controller identifies the connected LINAX® linear motor axis and configures the control parameters automatically. Each XENAX® Servo Controller can control one linear motor axis.



ELAX® Linear Motor Slide

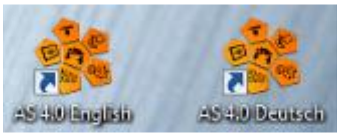
The ELAX® Linear Motor Slide are predestined for fast, precise positioning tasks. It's a Modular system with strokes of 30-150mm. The variable one-cable connection can be mounted on the back or sidewise. The XENAX® Servo Controller identifies the connected ELAX® and parametrizes automatically.



2 Automation Studio

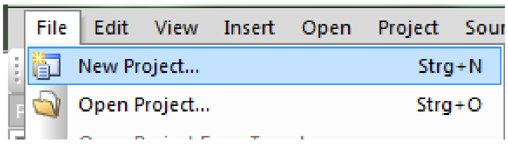
2.1 New Project

In the following steps it will be shown how a new project can be opened.



Start the Automation Studio from B&R.

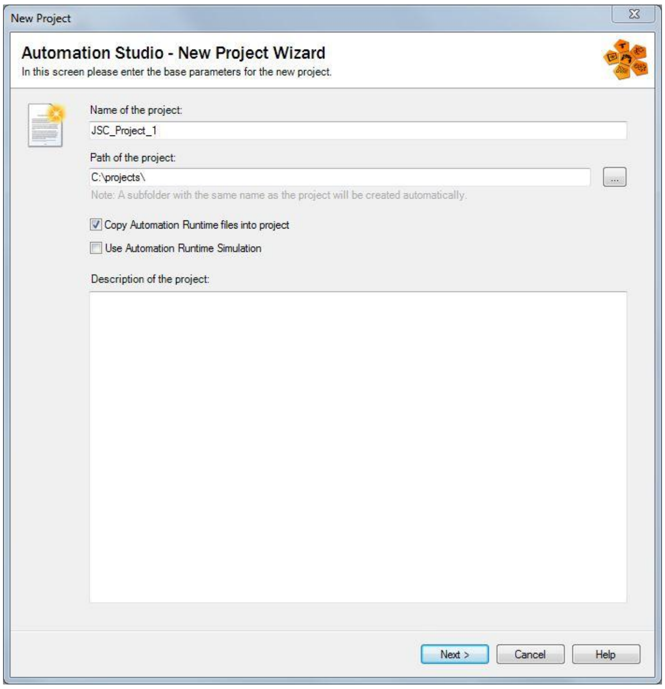
File -> New Projekt ->



Enter your project name e.g. „JSC_Project_1“
The storage path will be shown and can be changed.

Select
☑Copy Automation Runtime support.....

☐Use Automation Runtime Simulation.
It is not necessary to run a simulation if the PLC is used as hardware.



New Project

Automation Studio - New Project Wizard

In this screen please enter the base parameters for the new project.

Name of the project: JSC_Project_1

Path of the project: C:\projects\

Note: A subfolder with the same name as the project will be created automatically.

☑ Copy Automation Runtime files into project

☐ Use Automation Runtime Simulation

Description of the project:

-> Next

A Configuration name suggestion will be shown.

Our Configuration name is:
„Hardware_Config_1“

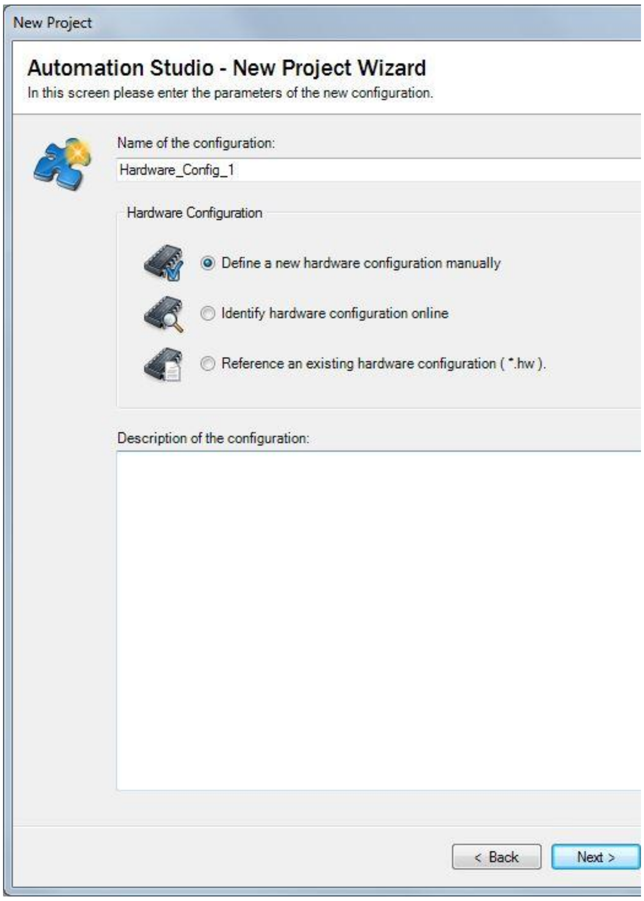
☑ **Define a new hardware configuration manually**

The basic hardware for the configuration can be assembled from the following pages of the wizard. This option has to be selected if Automation Studio is used for the first time and no B&R control is connected to the PC.

☐ **Identify hardware configuration online**
The hardware configuration will be loaded automatically from the connected B&R control The hardware configuration is automatically visible in the configuration.

☐ **Reference an existing hardware configuration**
The existing hardware configuration from another Automation Studio project or from another configuration in the same project can be used as a reference.

-> Next



New Project

Automation Studio - New Project Wizard

In this screen please enter the parameters of the new configuration.

Name of the configuration: Hardware_Config_1

Hardware Configuration

● Define a new hardware configuration manually

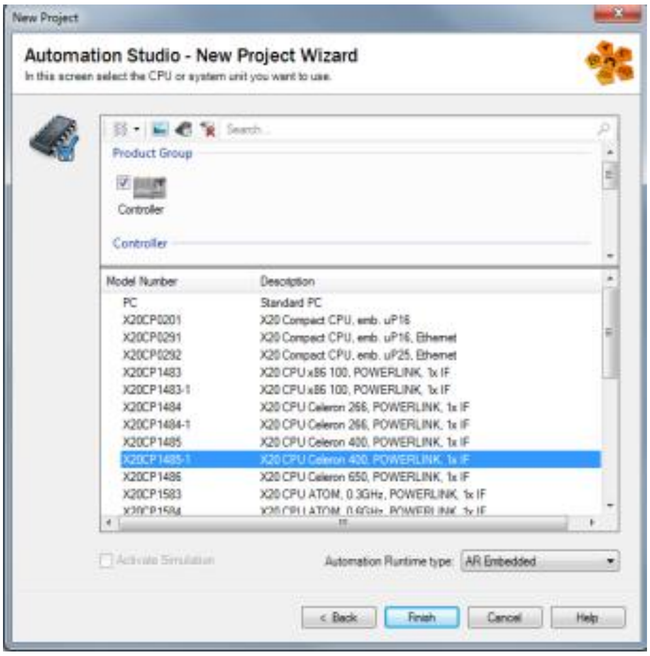
○ Identify hardware configuration online

○ Reference an existing hardware configuration (*.hw).

Description of the configuration:

We choose the B&R PLC X20CP1485-1

-> Finish



New Project

Automation Studio - New Project Wizard

In this screen select the CPU or system unit you want to use.

Product Group

☑ Controller

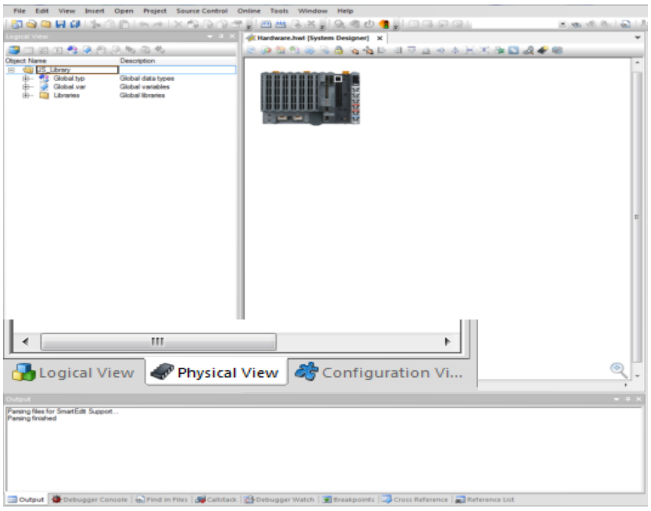
Controller

Model Number	Description
PC	Standard PC
X20CP0201	X20 Compact CPU, emb. uP16
X20CP0291	X20 Compact CPU, emb. uP16, Ethernet
X20CP0292	X20 Compact CPU, emb. uP25, Ethernet
X20CP1483	X20 CPU x86 100, POWERLINK, 1x IF
X20CP1483-1	X20 CPU x86 100, POWERLINK, 1x IF
X20CP1484	X20 CPU Celeron 266, POWERLINK, 1x IF
X20CP1484-1	X20 CPU Celeron 266, POWERLINK, 1x IF
X20CP1485	X20 CPU Celeron 400, POWERLINK, 1x IF
X20CP1485-1	X20 CPU Celeron 400, POWERLINK, 1x IF
X20CP1486	X20 CPU Celeron 650, POWERLINK, 1x IF
X20CP1583	X20 CPU ATOM, 0.3GHz, POWERLINK, 1x IF
X20CP1584	X20 CPU ATOM, 0.6GHz, POWERLINK, 1x IF

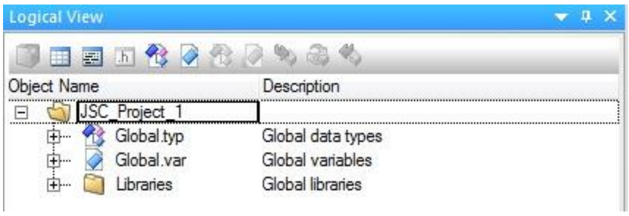
☐ Activate Simulation

Automation Runtime type: AR Embedded

You can choose between three different views: ***Logical View, Physical View*** and ***Configuration View***. After including the B&R control, each view has now basic settings.

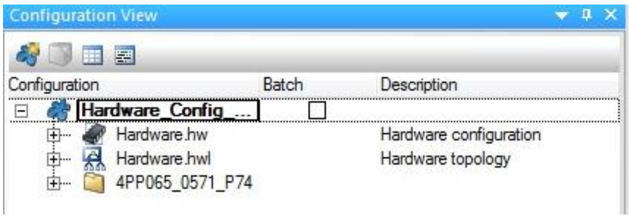


The **Logical View** represents the hardware independent view of the application. Use this view to list and manage data type declarations, variable declarations, packages, programs, libraries, data objects and documentation files.



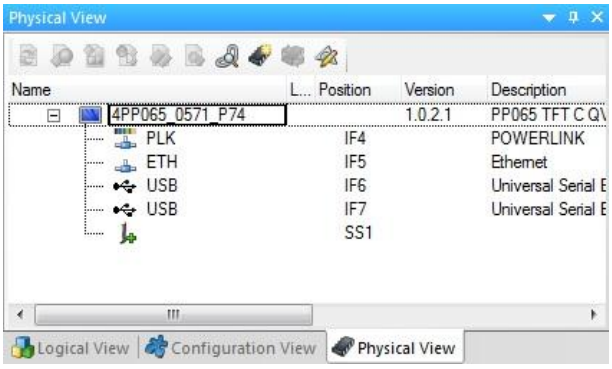
Object Name	Description
JSC_Project_1	
Global.typ	Global data types
Global.var	Global variables
Libraries	Global libraries

The **Configuration View** represents the hardware dependent application perspective. Use this view to display and manage the hardware configuration, software configuration files, declaration of permanent variables, I/O routings, Automation Runtime configuration, configuration NC, NC routings, VC keyboard mapping, configuration-specific data objects and documentation files.



Configuration	Batch	Description
Hardware_Config_...	□	
Hardware.hw		Hardware configuration
Hardware.hwl		Hardware topology
4PP065_0571_P74		

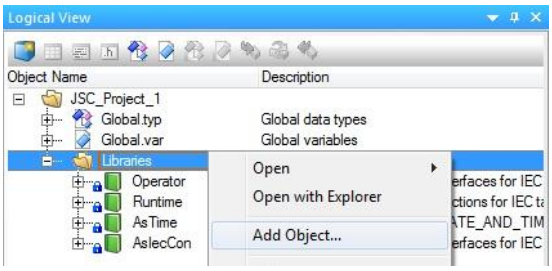
The **Physical View** provides the hardware point view of the active configuration. It displays the active configuration hardware modules in a directory tree.



Name	L...	Position	Version	Description
4PP065_0571_P74			1.0.2.1	PP065 TFT C Q\
PLK		IF4		POWERLINK
ETH		IF5		Ethernet
USB		IF6		Universal Serial E
USB		IF7		Universal Serial E
		SS1		

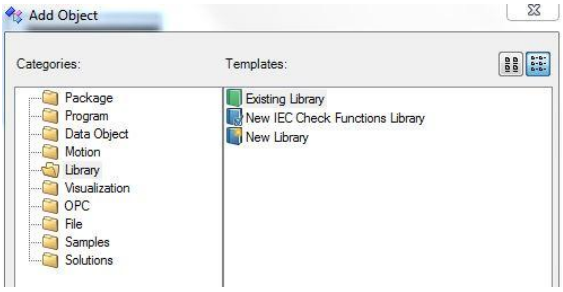
2.2 Import of the JsMcLib library

In the *Logical View* right click on the folder Libraries,
-> Add Object…



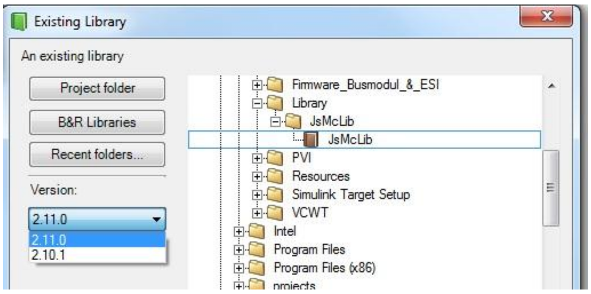
Categories -> Library -> Existing Library

-> Next



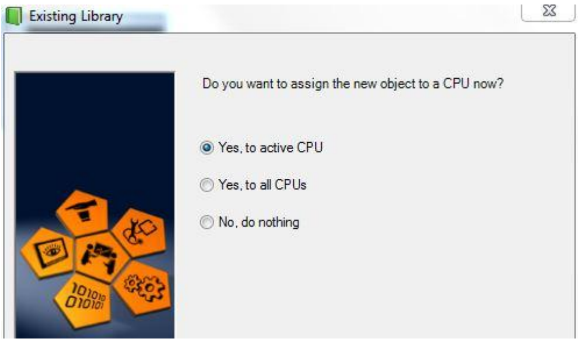
In the following step you have to select the "JsMcLib" library. If there are several versions, you can select the latest version in the "Version" drop-down list on the left side.

-> Next



With **"Yes, to active CPU"** the "library" is added to the active "CPU".
By choosing **"Yes, to all CPUs**" the library is added to all CPU's, in the case of more than a configured CPU.
By choosing **"No, do nothing**", the "library" will not be assigned to any CPU. The assignment has to be done manually.

-> Finish



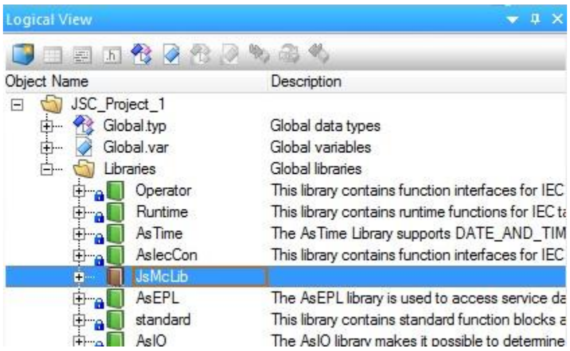
Do you want to assign the new object to a CPU now?

● Yes, to active CPU

○ Yes, to all CPUs

○ No, do nothing

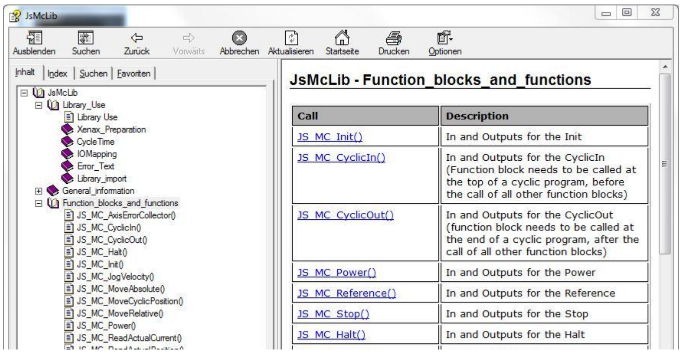
The inserted library is now visible in the *Logical View -> Libraries..*



Object Name	Description
JSC_Project_1	
Global.typ	Global data types
Global.var	Global variables
Libraries	Global libraries
Operator	This library contains function interfaces for IEC
Runtime	This library contains runtime functions for IEC ta
AsTime	The AsTime Library supports DATE_AND_TIM
AslecCon	This library contains function interfaces for IEC
JsMcLib	
AsEPL	The AsEPL library is used to access service da
standard	This library contains standard function blocks a
AsIO	The AsIO library makes it possible to determine

2.2.1 JsMcLib User Help

The help for the function blocks of the JsMcLib library, can be opened by selecting the JsMcLib library and then press the “F1” button.



JsMcLib - Function_blocks_and_functions

Call	Description
JS_MC_Init()	In and Outputs for the Init
JS_MC_CyclicIn()	In and Outputs for the CyclicIn (Function block needs to be called at the top of a cyclic program, before the call of all other function blocks)
JS_MC_CyclicOut()	In and Outputs for the CyclicOut (function block needs to be called at the end of a cyclic program, after the call of all other function blocks)
JS_MC_Power()	In and Outputs for the Power
JS_MC_Reference()	In and Outputs for the Reference
JS_MC_Stop()	In and Outputs for the Stop
JS_MC_Halt()	In and Outputs for the Halt

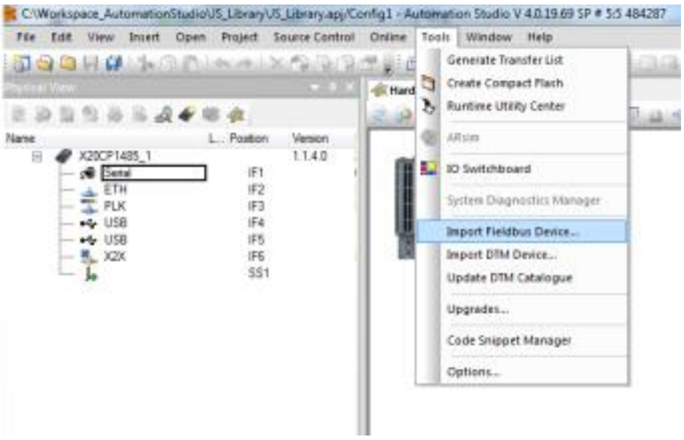
2.3 Embed XENAX® CANopen object file (.XDD)

The XENAX® Servo Controller is operating according to the device profile for electric drives CANopen DS402.
This device profile then goes on the Powerlink Ethernet fieldbus (CANopen over Powerlink).

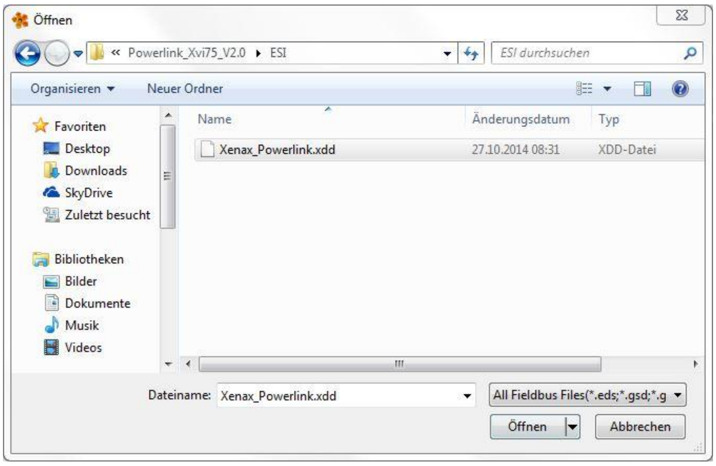
T

The CANopen object list is included in the description file “Xenax_Powerlink.xdd” and will be integrated into the Automation Studio.
Main menu -> *Tools -> Import Fieldbus Device….*

These objects are provided in readable form by the “Xenax_Powerlink.html” file on the delivered CD under *XENAX® _Xvi_75V8 \ XENAX® _Xvi_Powerlink\ or on* www.jennyscience.de/en/.



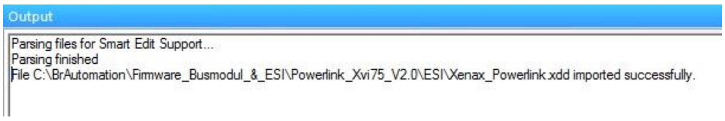
Choose the path of the .xdd-File -> *Open.*



Name	Änderungsdatum	Typ
Xenax_Powerlink.xdd	27.10.2014 08:31	XDD-Datei

Dateiname: Xenax_Powerlink.xdd
All Fieldbus Files(*.eds;*.gsd;*.g

„… imported successfully“ will be shown on the „Output“ window

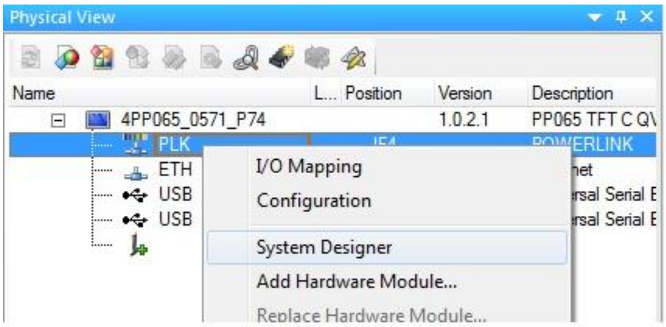


```
Parsing files for Smart Edit Support...
Parsing finished
File C:\BrAutomation\Firmware_Busmodul_&_ESI\Powerlink_Xvi75_V2.0\ESI\Xenax_Powerlink.xdd imported successfully.
```

2.4 Embed XEANAX® Servo Controller in project

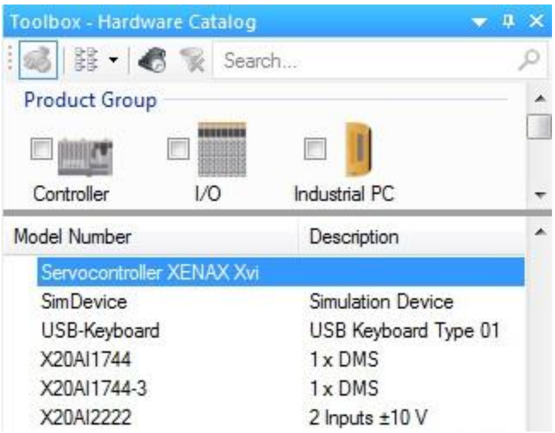
Include „Servo Controller XENAX Xvi“ from Toolbox into the project

Physical View-> select the Power Link Interface -> right click -> System Designer.



Name	L...	Position	Version	Description
4PP065_0571_P74			1.0.2.1	PP065 TFT C Q\
PLK		IF4		POWERLINK
ETH				net
USB				rsal Serial E
USB				rsal Serial E

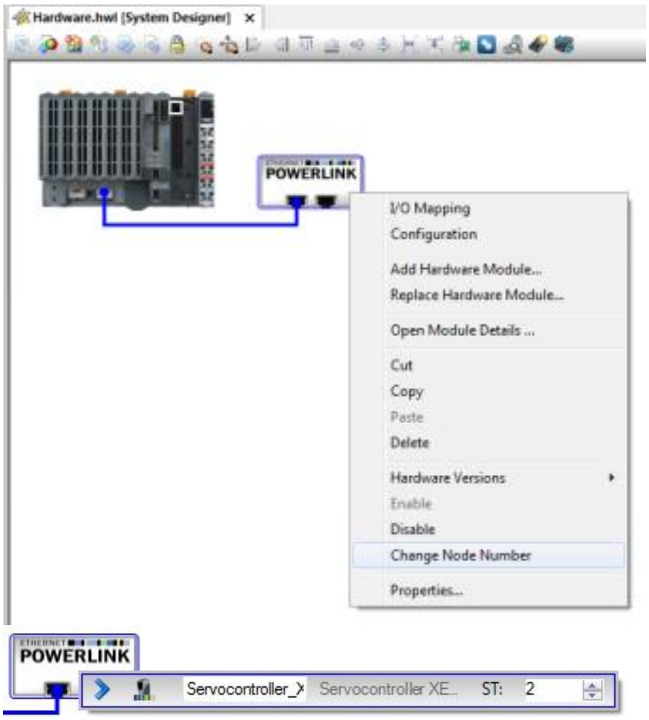
Toolbox-> Hardware Catalog select “Servo Controller XENAX Xvi” and doubleclick it to include to the System Designer.



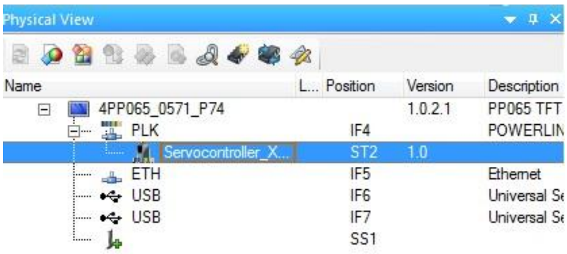
Model Number	Description
Servocontroller XENAX Xvi	
SimDevice	Simulation Device
USB-Keyboard	USB Keyboard Type 01
X20AI1744	1 x DMS
X20AI1744-3	1 x DMS
X20AI2222	2 Inputs ±10 V

For the connection, the „Node Number” hast to be the same as the “Card Identifier” (CI) on the XENAX® Servo Controller.

The CI number can easily be changed on the WebMotion® under “*by command line” with the ASCII code e.g. “CI2”*
Read with “CI?”



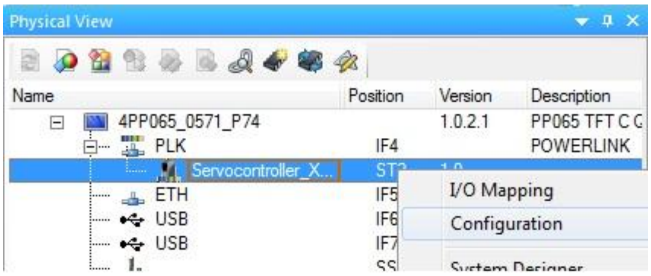
The Servo Controller XENAX® Xvi75V8 can now be seen in the Physical View under PLK(Powerlink).


Physical View

Name	L...	Position	Version	Description
4PP065_0571_P74			1.0.2.1	PP065 TFT
PLK		IF4		POWERLIN
Servocontroller_X...		ST2	1.0	
ETH		IF5		Ethernet
USB		IF6		Universal Se
USB		IF7		Universal Se
		SS1		

2.4.1 Configuration XENAX® based on the xdd description

Physical View/PLK/Servocontroller_XENAX_Xvi -> right click -> Configuration.



Physical View

Name	Position	Version	Description
4PP065_0571_P74		1.0.2.1	PP065 TFT C C
PLK	IF4		POWERLINK
Servocontroller_X...	ST		
ETH	IF5		
USB	IF6		
USB	IF7		
	SS		

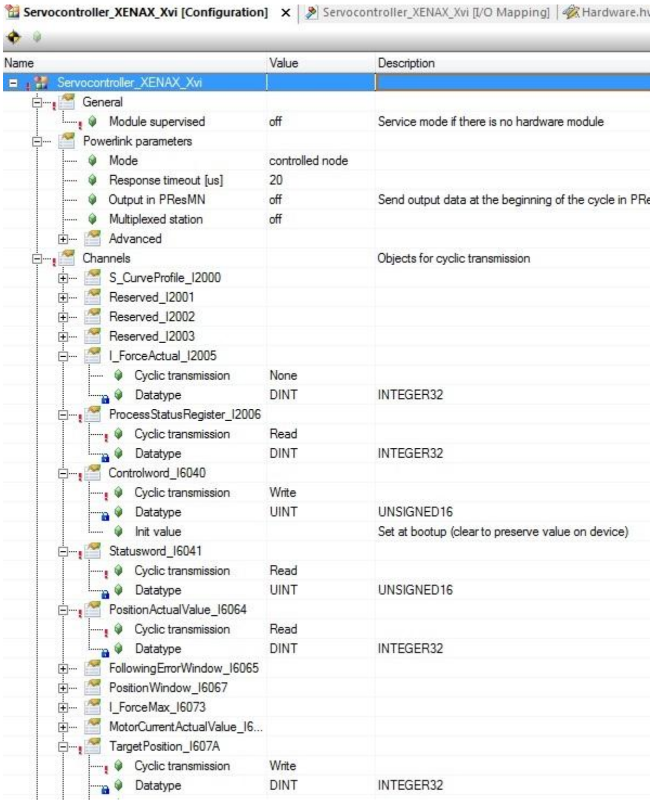
2.4.1.1 Cyclic transmission

In the submenu " Channels" all usable objects will be shown.

Important:
The Objects 1001h (Error bit register) and 1F8Ch (NMT_CurrNMTState,-> Network Managment status) are not mapable as PDO. This information should not unnecessarily burden the real-time cycle.

Objects which are transferred cyclic, can be activated.

Fort the JsMcLib library the following objects have to be transferred cyclic in order that the "**Cyclic Synchronized Mode"** can be used.



Servocontroller_XENAX_Xvi [Configuration]
Servocontroller_XENAX_Xvi [I/O Mapping]
Hardware.hw

Name	Value	Description
Servocontroller_XENAX_Xvi		
General		
Module supervised	off	Service mode if there is no hardware module
Powerlink parameters		
Mode	controlled node	
Response timeout [us]	20	
Output in PResMN	off	Send output data at the beginning of the cycle in PRe
Multiplexed station	off	
Advanced		
Channels		Objects for cyclic transmission
S_CurveProfile_I2000		
Reserved_I2001		
Reserved_I2002		
Reserved_I2003		
I_ForceActual_I2005		
Cyclic transmission	None	
Datatype	DINT	INTEGER32
ProcessStatusRegister_I2006		
Cyclic transmission	Read	
Datatype	DINT	INTEGER32
Controlword_I6040		
Cyclic transmission	Write	
Datatype	UINT	UNSIGNED16
Init value		Set at bootup (clear to preserve value on device)
Statusword_I6041		
Cyclic transmission	Read	
Datatype	UINT	UNSIGNED16
PositionActualValue_I6064		
Cyclic transmission	Read	
Datatype	DINT	INTEGER32
FollowingErrorWindow_I6065		
PositionWindow_I6067		
I_ForceMax_I6073		
MotorCurrentActualValue_I6...		
TargetPosition_I607A		
Cyclic transmission	Write	
Datatype	DINT	INTEGER32

Name of the Object	**Mode of transmission**
ProcessStatusRegister_I2006	Read
Controlword_I6040Out	Write
Statusword_I6041	Read
PositionActualValue_I6064	Read
TargetPosition_I607AOut	Write

If the **„Profile Position Mode"** is used, the following object have to be transferred additionally.

Name of the Object	Mode of transmission
S_CurveProfile_I2000Out	Write
ProfileVelocity_I6081Out	Write
ProfileAcceleration_I6083Out	Write

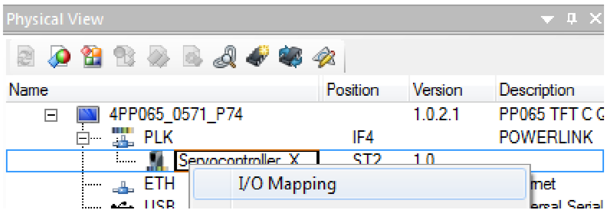
A detailed description of the Objects can be found in the document *Xvi75V8_CANopen_Ethernet* at page 17.
The document can be found on our website:
www.jennyscience.ch/downloads

Note:
If there are several XENAX® Servo Controller used, the object settings can easily be copied:

1. Select all object settings with „Ctrl"+ „A".
2. Right click-> *Copy*
3. *Open I/O Configuration* of the second Servo Controller select all objects with „Ctrl"+ „A".
4. Right click -> paste

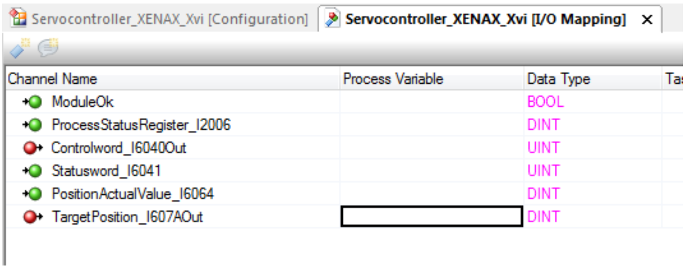
2.4.2 View I/O Mapping

The objects will be visible under the *I/O Mapping (Physical View)*.
In case the objects can't be seen, please check the settings of the *I/O Configuration.*



Name	Position	Version	Description
4PP065_0571_P74		1.0.2.1	PP065 TFT C G
PLK	IF4		POWERLINK
Servocontroller_X	ST2	1.0	
ETH			net
USB			ernal Serial

The I/O mapping should look like displayed on the right
.


Servocontroller_XENAX_Xvi [Configuration]
Servocontroller_XENAX_Xvi [I/O Mapping]

Channel Name	Process Variable	Data Type	Ta
ModuleOk		BOOL	
ProcessStatusRegister_I2006		DINT	
Controlword_I6040Out		UINT	
Statusword_I6041		UINT	
PositionActualValue_I6064		DINT	
TargetPosition_I607AOut		DINT	

3 JsMcLib

3.1 State Diagram of the Function Blocks

The following diagram shows the state and the behaviour of the axis when multiple motion control function blocks are “simultaneously” active.

Each motion command is a transition that changes the state of the axis and, as a consequence, influences the method of calculation of the current movement.

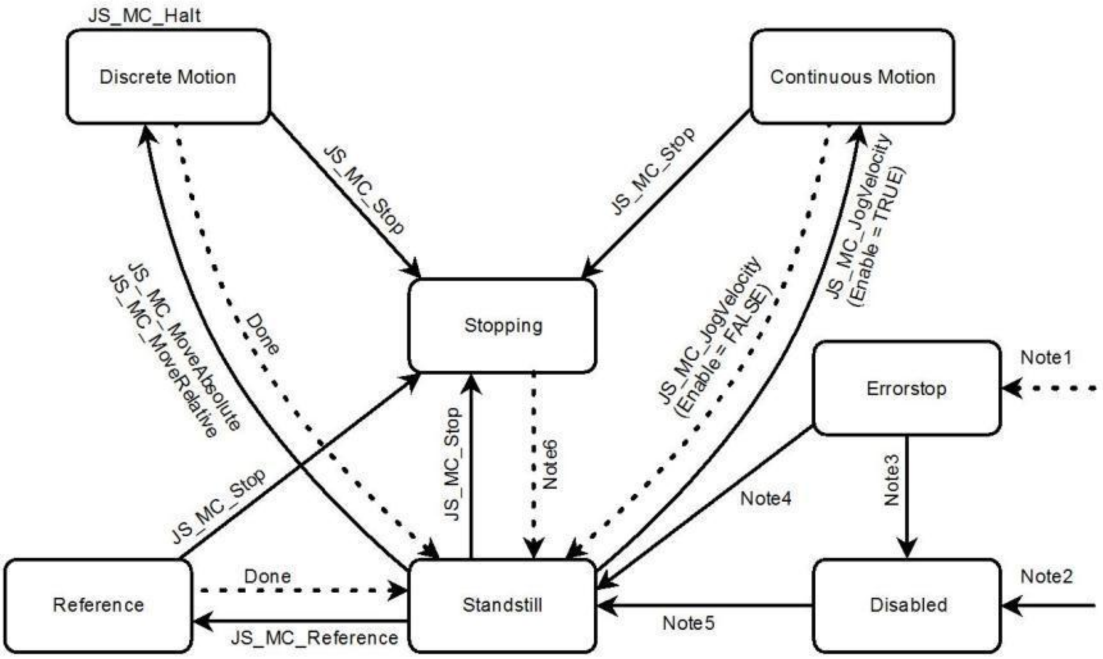
All function blocks which do not appear in the state diagram, do not affect the state of the axis.

The current state of the axis can be determined with the function block **“JS_MC_ReadStatus”**. If a function block is called where it is not allowed, the function block reports an error.

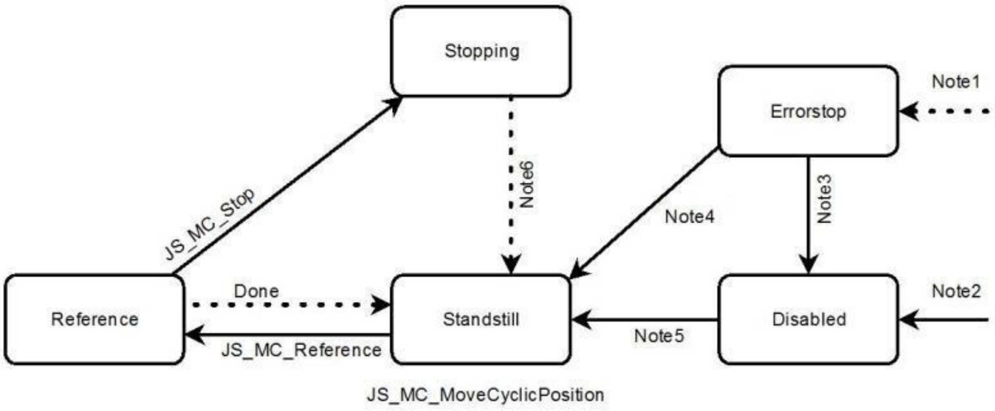
The notes describe the necessary conditions that must be met for a change in an axis state.

Important:
In the states “**Stopping**”, “**ErrorStop**”, ”**Disabled**” and “**Reference**” no motion blocks can be called. In standstill condition, an axis must always be referenced before starting a movement.

3.1.1 Profile Position Mode



3.1.2 Cyclic Synchronized Mode



In the Cyclic Synchronized mode the JS_MC_ReadStatus will be in „Standstill" after the reference.
To know which motion-commands can be used, the "MC_ReadStatus" of the virtual Axis has to be executed.

Note 1:
From any state. An error in the axis occurred.

Note 2:
From any state. JS_MC_Power.Enable = FALSE and there is no error in the axis.

Note 3:
JS_MC_Reset AND JS_MC_Power.Status = FALSE.

Note 4:
JS_MC_Reset AND JS_MC_Power.Status = TRUE AND JS_MC_Power.Enable = TRUE

Note 5:
JS_MC_Power.Enable = TRUE AND JS_MC_Power.Status = TRUE

Note 6:
JS_MC_Stop.Done = TRUE AND JS_MC_Stop.Execute = FALSE

3.2 JsMcLib Operation Mode

To operate a XENAX® Servo Controller via Powerlink, we provide a library in the Automation Studio with the name JsMcLib (Jenny Science Motion Control Library). Below, you find an overview of the sequences for the two different coupling types, Profile Position Mode and Cyclic Synchronise Mode for which the JsMcLib is required as well as a description of the JsMcLib application.

Profile Position Mode:
The „Profile Position” Mode, is a point to point connection. The trajectory will be calculated by the XENAX® itself. The following parameters have to be set:

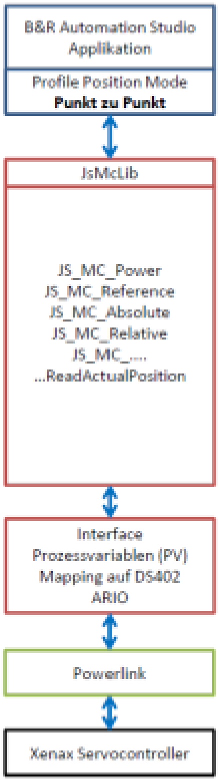
Endposition
Speed
Acceleration
S-Curve

Cyclic Synchronous Position Mode:
The „Cyclic Synchronous Position” Mode, is an interpolated drive.
The trajectory is calculated by the B&R PLC.
The only parameter is the position.
For this Mode a virtual B&R axis will be necessary.

XENAX® Profile Position Mode

Process:

1. Configure XENAX® Servo Controller with the Automation Studio
2. Power on with “**JS_MC_Power**”.
3. Reference XENAX® Servo Controller with “**JS_MC_Reference**”.
4. XENAX® Servo Controller is now ready to move e.g. an absolute way with the function block “**JS_MC_MoveAbsolute**”.



XENAX® Cyclic Synchronous Position Mode

Process:

1. Configure from the XENAX® Servo Controller and the virtual ACOPOS axis from B&R.

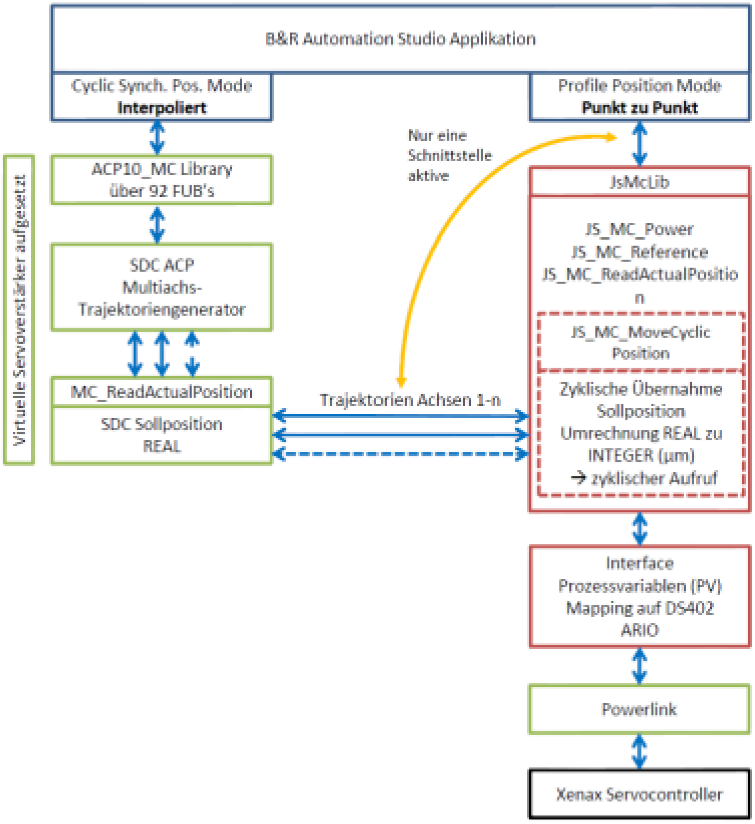
2. Power on XENAX® Servo Controller with "**JS_MC_Power**" and the virtual axis with "**MC_Power on**".

3. Reference XENAX® Servo Controller with "**JS_MC_Reference**" and prepare for the coupling to the virtual axis.

4. Reference virtual axis with "**MC_Home**" and then define the current position of the XENAX® Servo Controller as "HomePosition".

5. The XENAX® Servo Controller and the virtual axis are coupled together by attaching the calculated position of the virtual axis to the input "Position" of the function block "**JS_MC_MoveCyclicPosition**".

Now all the features of the virtual axis are available. The XENAX® Servo Controller now behaves like an ACOPOS axis from B&R.


B&R Automation Studio Applikation
Cyclic Synch. Pos. Mode
Interpoliert
Profile Position Mode
Punkt zu Punkt
Nur eine Schnittstelle aktive
Virtuelle Servoverstärker aufgesetzt
ACP10_MC Library über 92 FUB's
SDC ACP Multiachs-Trajektoriengenerator
MC_ReadActualPosition
SDC Sollposition REAL
Trajektorien Achsen 1-n
JsMcLib
JS_MC_Power
JS_MC_Reference
JS_MC_ReadActualPositio n
JS_MC_MoveCyclic Position
Zyklische Übernahme Sollposition Umrechnung REAL zu INTEGER (µm)
→ zyklischer Aufruf
Interface Prozessvariablen (PV) Mapping auf DS402 ARIO
Powerlink
Xenax Servocontroller

3.3 Function blocks of the JsMcLib

The function blocks of the JsMcLib intend to simplify the use of XENAX® Servo Controllers in the Automation Studio.
The function blocks of the JsMcLib are described below.

Function blocks:

Output stage power on/ off

	JS_MC_Power		
UDINT	Axis	Status	BOOL
BOOL	Enable	Valid	BOOL
		Error	BOOL
		ErrorID	UINT

Stops the active movement and changes to the “Stopping” state (The output stage turns off).

	JS_MC_Stop		
UDINT	Axis	Done	BOOL
BOOL	Execute	Busy	BOOL
UDINT	Deceleration	CommandAborted	BOOL
		Error	BOOL
		ErrorID	UINT

Stops the active movement and changes to the “Standstill” state (The output stage remains active).

	JS_MC_Halt		
UDINT	Axis	Done	BOOL
BOOL	Execute	Busy	BOOL
UDINT	Deceleration	CommandAborted	BOOL
		Error	BOOL
		ErrorID	UINT

Deletes an error on the XENAX® Servo Controller.

	JS_MC_Reset		
UDINT	Axis	Done	BOOL
BOOL	Execute	Busy	BOOL
		Error	BOOL
		ErrorID	UINT

Drive to an absolute position.

	JS_MC_MoveAbsolute		
UDINT	Axis	Done	BOOL
BOOL	Execute	Busy	BOOL
DINT	Position	CommandAborted	BOOL
UDINT	Velocity	Error	BOOL
UDINT	Acceleration	ErrorID	UINT
UDINT	Scurve		

		JS_MC_MoveRelative		
Drive a relative distance.	UDINT	Axis	Done	BOOL
	BOOL	Execute	Busy	BOOL
	DINT	Distance	CommandAborted	BOOL
	UDINT	Velocity	Error	BOOL
	UDINT	Acceleration	ErrorID	UINT
	UDINT	Scurve		

		JS_MC_JogVelocity		
Driving with a pre-set velocity, as long	UDINT	Axis	Active	BOOL
as “JogPositive” or “JogNegative” is	BOOL	Enable	Busy	BOOL
active				
And the end / soft limit is not reached.	UDINT	Velocity	CommandAborted	BOOL
	UDINT	Acceleration	Error	BOOL
	UDINT	Deceleration	ErrorID	UINT
	BOOL	JogPositive	Jogging	BOOL
	BOOL	JogNegative		

		JS_MC_MoveCyclicPosition		
Coupling the XENAX® Servo Controller	UDINT	Axis	Valid	BOOL
with the virtual B&R axis. Motion	BOOL	Enable	CommandAborted	BOOL
commands are triggered by B&R Axis.	DINT	Position	Error	BOOL
			ErrorID	UINT

		JS_MC_ReadStatus		
Read status of the XENAX® Servo	UDINT	Axis	Valid	BOOL
Controller.	BOOL	Enable	Error	BOOL
			ErrorID	UINT
			Errorstop	BOOL
			Disabled	BOOL
			Stopping	BOOL
			Standstill	BOOL
			DiscreteMotion	BOOL
			Reference	BOOL
			ContinuousMotion	BOOL

		JS_MC_ReadActualPosition		
Read actual position.	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Error	BOOL
			ErrorID	UINT
			Position	DINT

		JS_MC_ReadAxisError		
Read axis error from XENAX® Servo Controller.	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Busy	BOOL
	BOOL	Acknowledge	Error	BOOL
	UDINT	DataAddress	ErrorID	UINT
	UINT	DataLength	ErrorRecordAvailable	BOOL
	STRING[12]	DataObjectName	ErrorRecord	JS_MC_Error Record
			FunctionBlockErrorCount	UINT
			AxisErrorCount	UINT
			AxisWarningCount	UINT

		JS_MC_ReadDigitalInput		
Read digital inputs from XENAX® Servo Controller	UDINT	Axis	Valid	Axis
	BOOL	Enable	Error	Enable
			ErrorID	
			DigitalInput	

		JS_MC_ReadDigitalOutput		
Read digital outputs from XENAX® Servo Controller	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Error	BOOL
			ErrorID	UINT
			DigitalOutput	USINT

		JS_MC_ReadParameter		
Read parameter values from XENAX® Servo Controller	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Error	BOOL
	INT	ParameterNumber	ErrorID	UINT
			Value	UDINT

		JS_MC_WriteParameter		
Write parameter value to the XENAX® Servo Controller.	UDINT	Axis	Done	BOOL
	BOOL	Execute	Error	BOOL
	INT	ParameterNumber	ErrorID	UINT
	DINT	Value		

		JS_MC_WriteDigitalOutput		
Write digital outputs for XENAX® Servo Controller.	UDINT	Axis	Done	BOOL
	BOOL	Execute	Error	BOOL
	USINT	DigitalOutput	ErrorID	UINT

* The Jenny Science specific objects are described in the document “Xvi75V8_CANopen_Ethernet” and can be found under www.jennyscience.de/en/.

		JS_MC_AxisErrorCollector		
Check if there is an axis or a function block error pending.	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Busy	BOOL
			Error	BOOL
			ErrorID	UINT
			FunctionBlockError	BOOL
			AxisError	BOOL
			AxisWarning	BOOL
			Errorstop	UINT

		JS_MC_ReadActualCurrent		
Read actual current of the connected linear motor axis.	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Error	BOOL
			ErrorID	UINT
			Current	INT

		JS_MC_ReadPSR		
Read Jenny Science process status register.	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Error	BOOL
			ErrorID	UINT
			ProcessStatusRegister	JS_MC_Init_IS_ProcStat

		JS_MC_Reference		
Execute the reference of the linear motor.	UDINT	Axis	Done	BOOL
	BOOL	Execute	Busy	BOOL
	USINT	ReferenceMode	CommandAborted	BOOL
	UDINT	ReferenceSpeedRot	Error	BOOL
	UDINT	ZMarkSpeedRot	ErrorID	UINT

		JS_MC_Init		
In the init code, this function block has to be called once for every connected axis.	JS_MC_IoMap	AdrIoMap	Axis	UDINT
	UDINT	pDevice		
	USINT	Node		
	SINT	OperationMode		

		JS_MC_CyclicIn		
This function block has to be called at the beginning of the synchronous cyclic mode to get the actual values of the Powerlink fieldbus.	UDINT	Axis	Valid	BOOL
	BOOL	Enable	Error	BOOL
			ErrorID	UINT

		JS_MC_CyclicOut		
This function block need to be called at the end of a Cyclic move to send new position values to the Powerlink fieldbus.	UDINT	Axis		

3.3.1 Minimum and Maximum Values of Function Blocks

Following minimum and maximum values of the function blocks should be adhered to.

name	datatype	min	max
Velocity linear	UDINT	10 inc/s	9000000 inc/s
Velocity rotative	UDINT	10 inc/s	100000000 inc/s
Deceleration	UDINT	2000 inc/s^2	1000000000 inc/s^2
Acceleration	UDINT	2000 inc/s^2	1000000000 inc/s^2
S-curve	UDINT	1 %	100 %

3.3.1 Error in the Function Block

The following ErrorIDs can be generated by the JsMcLib function blocks.

Value	**Name**	**Description**	**Correction**
0	ERR_OK	FUB executed correctly with no errors	None.
50000	jsmcERR_NIL_POINTER	No axis passed to FB	Ensure function block call only with correct axis passed.
50001	jsmcERR_DRIVE_NOT_READY	controller is not ready to switch on	Check controller for errors
50002	jsmcERR_DRIVE_SWITCHED_OFF	controller is switched off	Don't call function block when controller is switched off
50004	jsmcERR_REFERENCE_WRONG_METHOD	Reference method is not correct for the motor	Check documentation for allowed reference methods for the motor
50006	jsmcERR_ACCE_TO_SMALL	Acceleration is to small	Use larger acceleration (>=2000 inc/s)
50008	jsmcERR_SCURVE_NOT_IN_RANGE	Scurve is not in allowed range	Use Scurve in allowed range (1...100%)
50010	jsmcERR_SDO_COMM_FAILURE	Failure during SDO communication	Check power link connection to the Servo Controller
50011	jsmcERR_POWER_UP_FAILURE	Failure during power up sequence	Check Servo Controller for correct power supply
50012	jsmcERR_POWER_LOST	Power was turned off outside of JS_MC_Power control	Check and quit errors from other function blocks or axis, which caused the power off
50013	jsmcERR_WRONG_STATE_FOR_FB	The FB cannot be used in the current state	Check program to call FB's only in allowed states

50014	jsmcERR_WRONG_OP_MODE_FOR_FB	The FB cannot be used in the current mode of operation	Only use allowed FB's for the desired mode of operation (profile position or cyclic synchronized)
50015	jsmcERR_EXECUTION_ERROR	The FB failed during execution by an external error	Check and quit errors from other function blocks or axes, which caused the fault
50016	jsmcERR_BUFFER_TO_SMALL	The buffer for the error text string is to small	Put a pointer to a buffer for the error text string which size is at least 50 characters
50017	jsmcERR_TEXT_OBJ_NOT_FOUND	Error text object or function block text object not found	Enter correct name of the error text object and ensure, that the error text object (JsMcEtxDe/JsMcEtxEn) and the function block text object (JsMcFBtxEn) are present in the project
50018	jsmcERR_TEXT_READOUT_FAILURE	Error text or function block text could not be read successfully	Ensure that the error text object (JsMcEtxDe/JsMcEtxEn) and the function block text object (JsMcFBtxEn) are present in the project
50019	jsmcERR_WRONG_GENERAL_OP_MODE	general mode of operation not supported	Set a supported general mode of operation in JS_MC_Init (OperationMode = jsmcMODE_PROFILE_POSITION or jsmcMODE_CYCLIC_SYNC)
50020	jsmcERR_REF_SPEED_NOT_IN_RANGE	Reference speed for rotative motors is out of range	Use reference speed in allowed range (0...250000 inc/s)
50021	jsmcERR_ZMARK_SPEED_NOT_IN_RANGE	Z-Mark speed for rotative motors is out of range	Use Z-Mark speed in allowed range (0...100000 inc/s)
50022	jsmcERR_VELOCITY_NOT_IN_RANGE	Velocity is out of range	Use velocity in allowed range (10...9000000 inc/s for linear motor, 10...100000000 inc/s for rotative motor)
50023	jsmcERR_ACCE_TO_LARGE	Acceleration is to large	Use smaller acceleration (smaller than 1000000000 inc/s^2)
50024	jsmcERR_CYCLE_TIME_FAILURE	Cycle time setting failure	Use correct cycle time setting (powerlink bus cycle time >= 400us and software task cycle time >= powerlink bus cycle time)
50025	jsmcERR_DECE_TO_SMALL	Deceleration is to small	Use larger deceleration (>=2000 inc/s)

50026	jsmcERR_DECE_TO_LARGE	Deceleration is to large	Use smaller deceleration (smaller than 1000000000 inc/s^2)
50027	jsmcERR_FW_VERS_FAILURE	Firmware version failure	For library use, ~~at least~~ XENAX firmware V3.64D and powerlink bus module firmware V2.0 or higher are required
50028	jsmcERR_PDO_MAPPING_CHK_FAILURE	Failure during PDO mapping check	Error in AsIOPVInfo() function block of AsIO library
50029	jsmcERR_PDO_MAPPING_MISSING	Necessary PDO mapping missing	Check, if all necessary PDOs are mapped in I/O Mapping
50030	jsmcERR_NO_DATA_ADDRESS_ASSIGNED	No data address for error text string assigned	Assign valid data address for error text string
50031	jsmcERR_SDO_ACCESS_FAILURE	Invalid SDO access	Check input values DataObject, SubID and DataLength and set correct values
50032	jsmcERR_CYCLIC_COMM_INTERRUPTED	Cyclic communication interrupted	Don't enable power until JS_MC_CyclicIn is valid and ~~so~~ cyclic communication is running

4 Point to Point or Interpolated Position Mode

Point to Point, Profile Position Mode:
In the „Profile Position“ mode, the parameters of distance, velocity, acceleration and s-curve will be transmitted to the XENAX®. The trajectory for the drive will be calculated internally in the XENAX® Servo Controller.

Interpolated, Cyclic Synchronous Position Mode:
In the „Cyclic Synchronous Position“ mode the position value will be transmitted cyclically (e.g. every millisecond) to every connected XENAX® Servo Controller. In this mode a virtual axis (no hardware mapping) for every connected XENAX® Servo Controller is necessary.

4.1 Hardware Setup

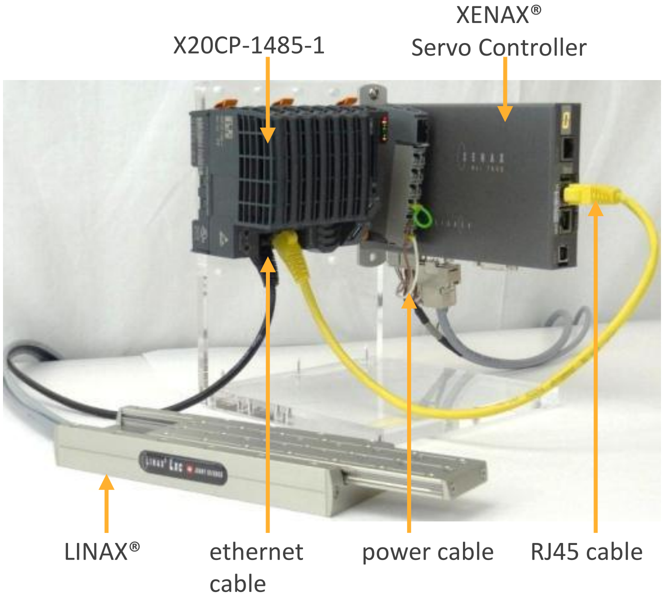
Using a XENAX® Servo Controller and a B&R PLC: X20CP1485-1

- XENAX® Xvi75V8 Servo Controller with additional Powerlink fieldbus module
- Include the XENAX® Xvi75V8 Servo Controller into a new project in Automation Studio.

4.2 Hardware Setup

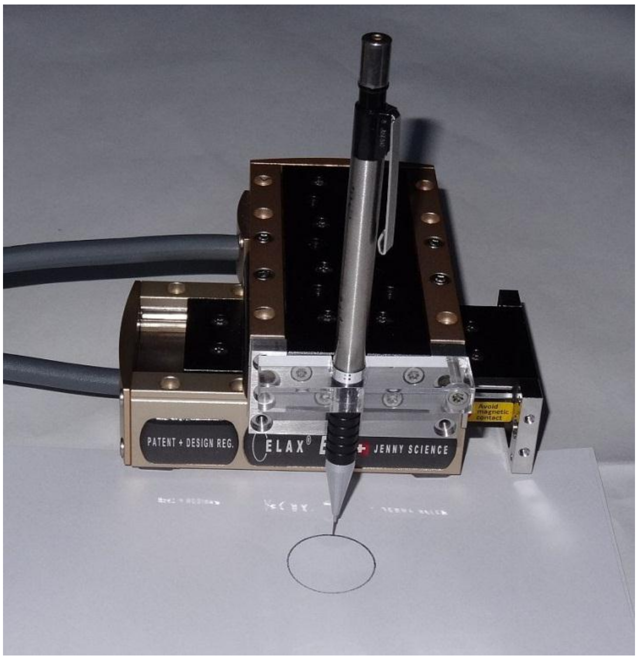
Profile Position Mode

1x B&R PLC X20CP1485-1
1x XENAX® Servo Controller with Powerlink fieldbus module.
1x Ethernet connection from B&R PLC to a PC
1x RJ45 cable for the connection of the B&R PLC and the XENAX® Servo Controller.
1x LINAX® Lxc 85F10 linear motor axis



Cyclic Synchronous Position Mode

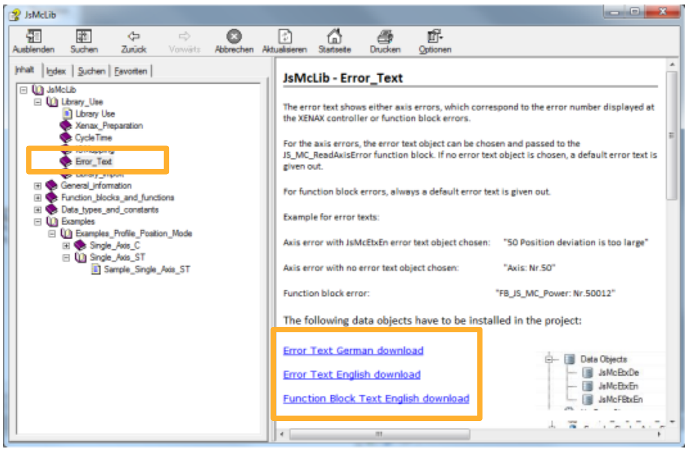
1 x B&R PLC X20CP1485-1
2x XENAX® Servo Controller with Powerlink fieldbus module
2x Ethernet connection from B&R PLC to a PC
2x RJ45 cable for the connection of the B&R PLC and the XENAX® Servo Controller.
2x ELAX® linear motor slide as X-Y cross table.



4.3 Embedding XENAX® Error Messages

Error on the XENAX® Servo Controller can be read with the „JS_MC_ReadAxosError” function block.
To provide this function block with an error text the “JsMcEtxEn” file has to be integrated into the Automation Studio.

Under JsMcLib ⊞ Library_Use -> Error_Text, the corresponding file in in German or English can be selected. Specify a desired location and extract the file by clicking right and press -> Extract All.



JsMcLib - Error_Text

The error text shows either axis errors, which correspond to the error number displayed at the XENAX controller or function block errors.

For the axis errors, the error text object can be chosen and passed to the JS_MC_ReadAxisError function block. If no error text object is chosen, a default error text is given out.

For function block errors, always a default error text is given out.

Example for error texts:

Axis error with JsMcEtxEn error text object chosen: "50 Position deviation is too large"

Axis error with no error text object chosen: "Axis: Nr.50"

Function block error: "FB_JS_MC_Power: Nr.50012"

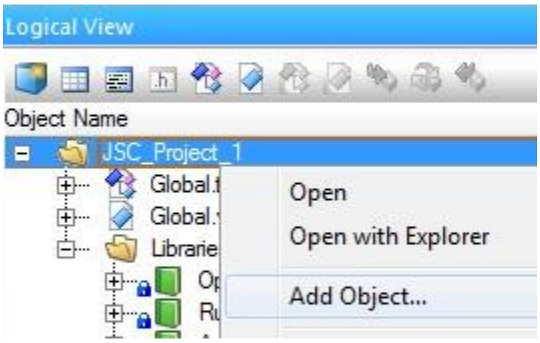
The following data objects have to be installed in the project:

Error Text German download

Error Text English download

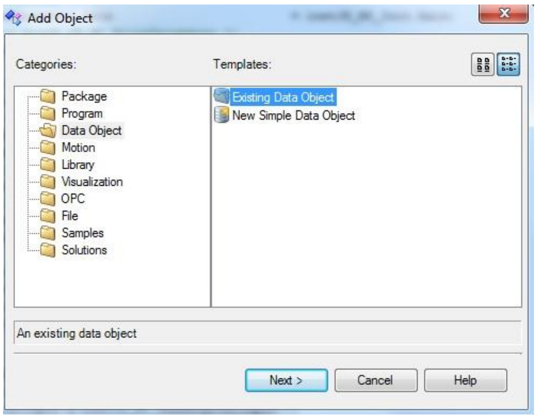
Function Block Text English download

In *Logical View select* the project name, „JSC_Project_1“,
-> Right click -> *Add Object*



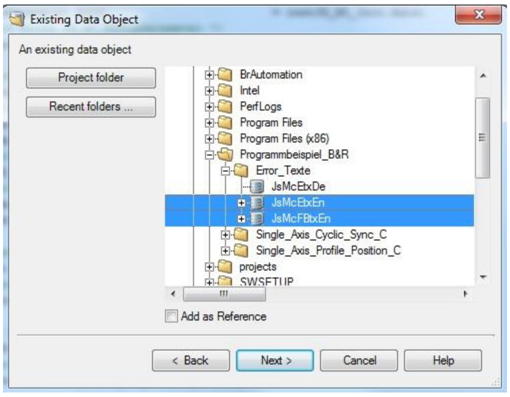
Categories: Data Object -> Templates: Existing Data Object

->Next



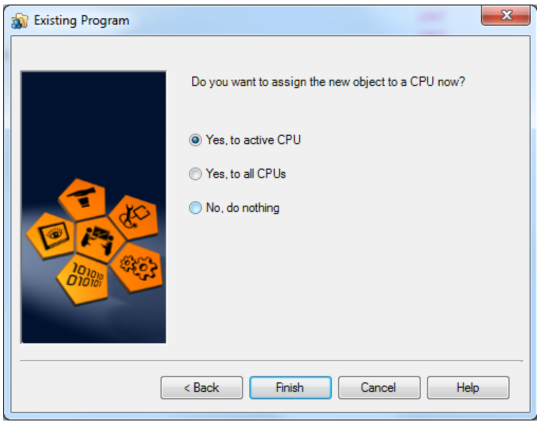
Select the file („JsMcEtxDe” or „JsMcEtxEn” and „JsMcFBtxEn“).

-> *Next*



Yes, to active CPU adds the “Error text” to the active CPU (here X20CP1485-1.).

-> *Finish*



Do you want to assign the new object to a CPU now?

● Yes, to active CPU

○ Yes, to all CPUs

○ No, do nothing

5 Program example “Profile Position Mode“

5.1 Get program example and save it

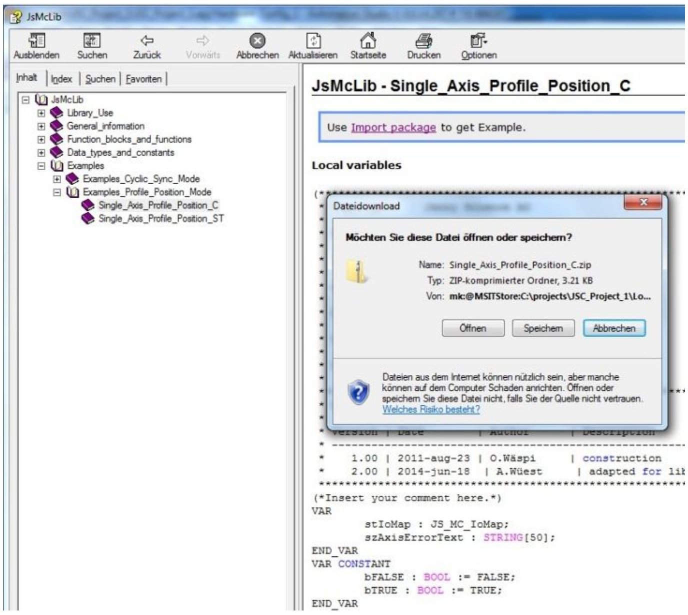
The program example is stored as c-coder or structured text in the “user help” menu of the JsMcLib.
Firstly the example needs to be saved to the PC and then it can be imported to the project.

Logical View -> select JsMcLib Library -> press F1 to open the user’s help.

Under Examples the program examples can be found.

In our Example we choose “Single_Axis_Profile_Position_C” and click on “Import Package”.

Then we save this file on the PC and extract the .zip-file. Right click -> “extract all”



JsMcLib - Single_Axis_Profile_Position_C

Use Import package to get Example.

Local variables

Möchten Sie diese Datei öffnen oder speichern?

Name: Single_Axis_Profile_Position_C.zip
Typ: ZIP-komprimierter Ordner, 3.21 KB
Von: mk:@MSITStore:C:\projects\JSC_Project_1\Lo...

Dateien aus dem Internet können nützlich sein, aber manche können auf dem Computer Schaden anrichten. Öffnen oder speichern Sie diese Datei nicht, falls Sie der Quelle nicht vertrauen. Welches Risiko besteht?

```
 *    1.00 | 2011-aug-23 | O.Wäspi      | construction
 *    2.00 | 2014-jun-18  | A.Wüest      | adapted for lib
 **************************************************************
(*Insert your comment here.*)
VAR
        stIoMap : JS_MC_IoMap;
        szAxisErrorText : STRING[50];
END_VAR
VAR CONSTANT
        bFALSE : BOOL := FALSE;
        bTRUE : BOOL := TRUE;
END_VAR
```

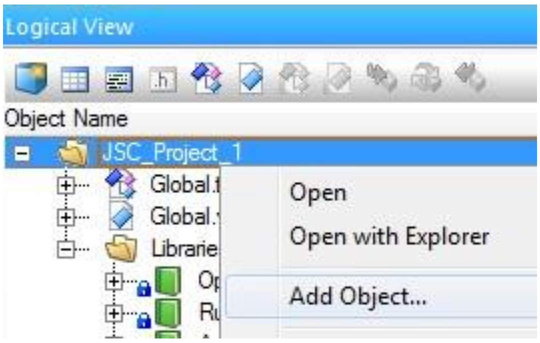
Note:

Program
A program has direct access to all variables, functions and function blocks.

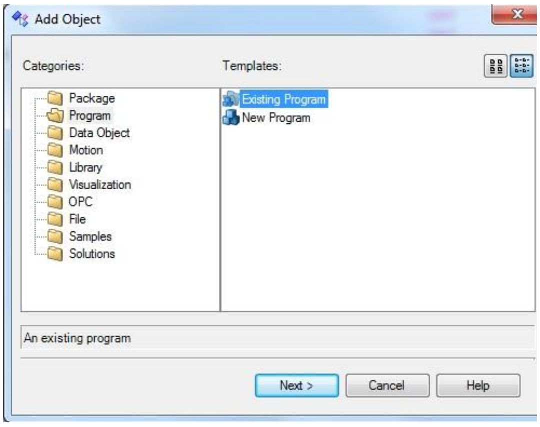
Package
Bigger Applications including a big amount of data objects, variables, functions and so on can be split into single units.
The summary of these units is called a „Package“ and can be named and imported.

5.2 Import program example to the project

Under *Logical View,* select the, „JSC_Project_1“.
-> Right click -> *Add Object*

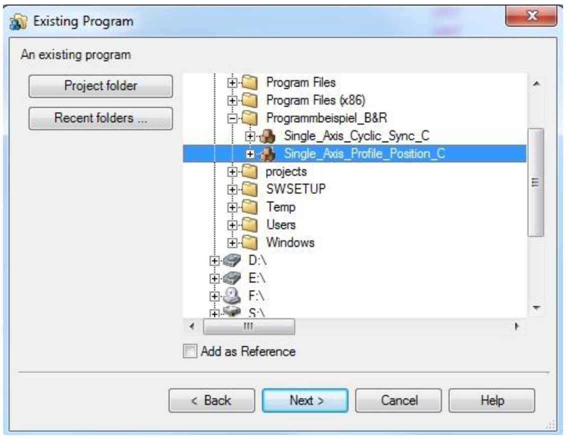


Categories -> Program -> Templates ->Existing Program



-> *Next*

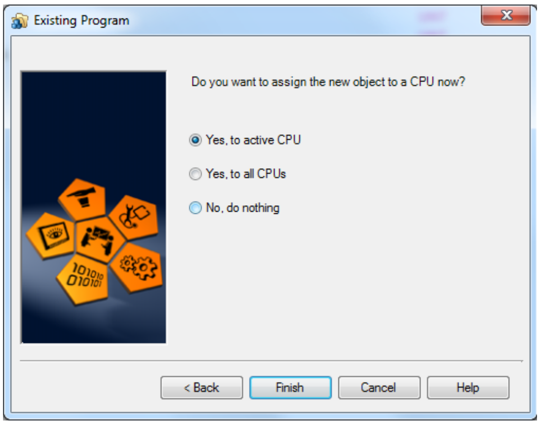
Select the program example, „Single_Axis_Profile_Position_C“.



□ Add as Reference

-> *Next*

Yes, to active CPU adds the program example to the active CPU (here X20CP1485-1.).



Do you want to assign the new object to a CPU now?

● Yes, to active CPU

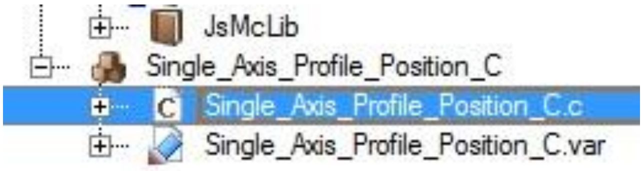
○ Yes, to all CPUs

○ No, do nothing

-> *Finish*

Check

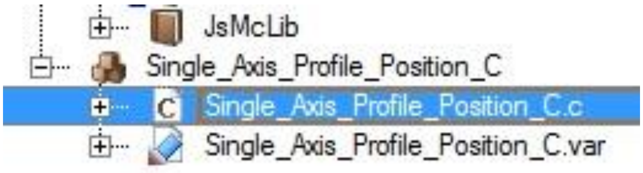
Under *Logical View* the *„Single_Axis_Profile_Position_C“* now appears.



JsMcLib
Single_Axis_Profile_Position_C
Single_Axis_Profile_Position_C.c
Single_Axis_Profile_Position_C.var

5.3 Configure Ethernet interface

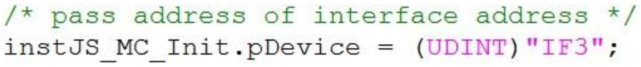
With double click on the “Single_Axis_Profile_Position_C.c” the program code is opened. The program code includes the Ethernet Powerlink interface address (IFx), Node number and the used error object.



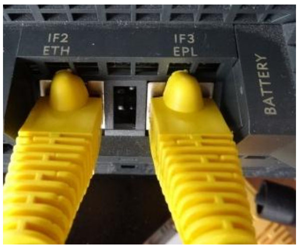
JsMcLib
Single_Axis_Profile_Position_C
Single_Axis_Profile_Position_C.c
Single_Axis_Profile_Position_C.var

Ethernet Powerlink address EPL:
The following line of the program example has to match the interface used.

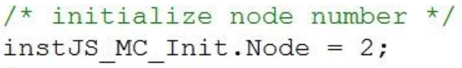
```
instJS_MC_Init.pDevice = (UDINT)”IF3”;
```



```
/* pass address of interface address */
instJS_MC_Init.pDevice = (UDINT)"IF3";
```



Node nr. Setting:
The Node number is set to 2 in the Line `instJS_MC_Init.Node= 2;`.
Set the Node number of the connected XENAX® Servo Controller (Node / Card Identifier CI, set with WebMotion®, see chapter “Card Identifier”).

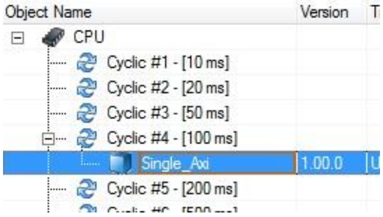


```
/* initialize node number */
instJS_MC_Init.Node = 2;
```

5.4 Move Program example into Task

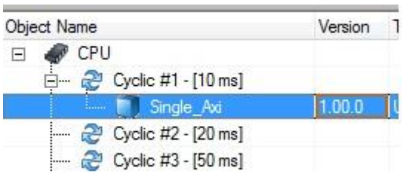
To see the *Software Configuration* go to *Physical View* and double click on B&R PLC (here *X20CP1485-1*).

The default cycle of the program is “Cyclic #4” (100ms).



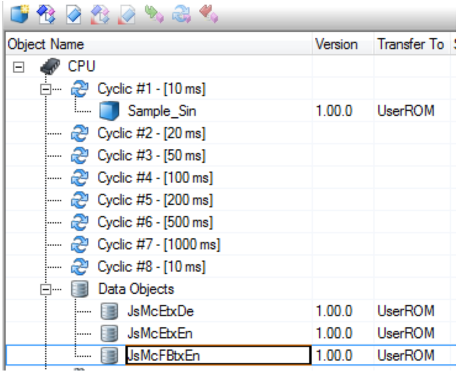
Object Name	Version	T
CPU		
Cyclic #1 - [10 ms]		
Cyclic #2 - [20 ms]		
Cyclic #3 - [50 ms]		
Cyclic #4 - [100 ms]		
Single_Axi	1.00.0	U
Cyclic #5 - [200 ms]		
Cyclic #6 - [500 ms]		

We move the program by drag & drop in “Cyclic #1” which has a shorter cycle time (10ms).



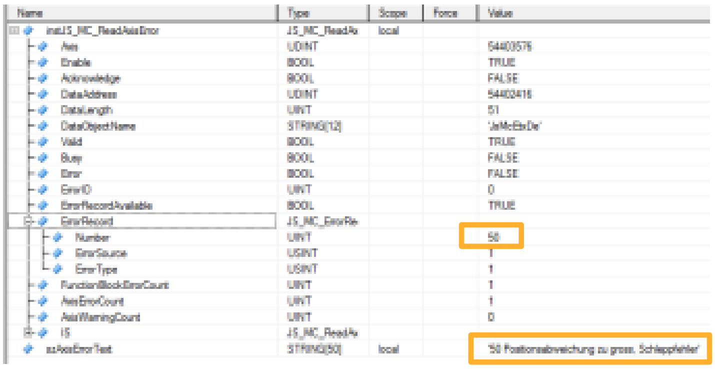
Object Name	Version	T
CPU		
Cyclic #1 - [10 ms]		
Single_Axi	1.00.0	U
Cyclic #2 - [20 ms]		
Cyclic #3 - [50 ms]		

To verify the correct integration, the error objects can be found in „Data Objects“.



Object Name	Version	Transfer To
CPU		
Cyclic #1 - [10 ms]		
Sample_Sin	1.00.0	UserROM
Cyclic #2 - [20 ms]		
Cyclic #3 - [50 ms]		
Cyclic #4 - [100 ms]		
Cyclic #5 - [200 ms]		
Cyclic #6 - [500 ms]		
Cyclic #7 - [1000 ms]		
Cyclic #8 - [10 ms]		
Data Objects		
JsMcEtxDe	1.00.0	UserROM
JsMcEtxEn	1.00.0	UserROM
JsMcFBtxEn	1.00.0	UserROM

Example:
If the XENAX® Servo Controller shows the Error 50 on the display, the function block “JS_MC_ReadAxisError” will show, “50 position deviation is too large”.

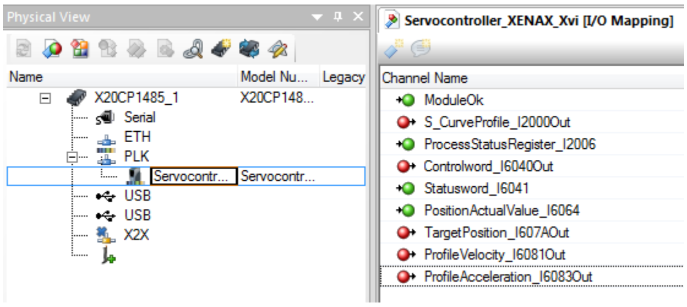


Name	Type	Scope	Force	Value
instJS_MC_ReadAxisError	JS_MC_ReadAx	local		
Axis	UDINT			54403576
Enable	BOOL			TRUE
Acknowledge	BOOL			FALSE
DataAddress	UDINT			54402416
DataLength	UINT			51
DataObjectName	STRING[12]			'JsMcEtxDe'
Valid	BOOL			TRUE
Busy	BOOL			FALSE
Error	BOOL			FALSE
ErrorID	UINT			0
ErrorRecordAvailable	BOOL			TRUE
ErrorRecord	JS_MC_ErrorRe			
Number	UINT			50
ErrorSource	USINT			1
ErrorType	USINT			1
FunctionBlockErrorCount	UINT			1
AxisErrorCount	UINT			1
AxisWarningCount	UINT			0
JS	JS_MC_ReadAx			
ssAxisErrorText	STRING[50]	local		'50 Positionsabweichung zu gross, Schleppfehler'

5.5 I/O Mapping, connect .xdd channels to program variables

To use the .xdd channels (cyclic data from the XENAX®) in the library, they have to be connected to the "Process Variable".

Physical View, double click XENAX® Servo Controller -> *I/O Mapping*.



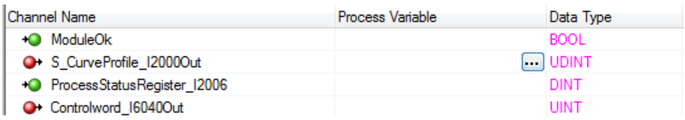
Physical View

Name	Model Nu...	Legacy
X20CP1485_1	X20CP148...	
Serial		
ETH		
PLK		
Servocontr...	Servocontr...	
USB		
USB		
X2X		

Servocontroller_XENAX_Xvi [I/O Mapping]

Channel Name
ModuleOk
S_CurveProfile_I2000Out
ProcessStatusRegister_I2006
Controlword_I6040Out
Statusword_I6041
PositionActualValue_I6064
TargetPosition_I607AOut
ProfileVelocity_I6081Out
ProfileAcceleration_I6083Out

Select the Channel on the row "Process Variable" (e.g. S_CurveProfil_I2000Out). Click on the … symbol.



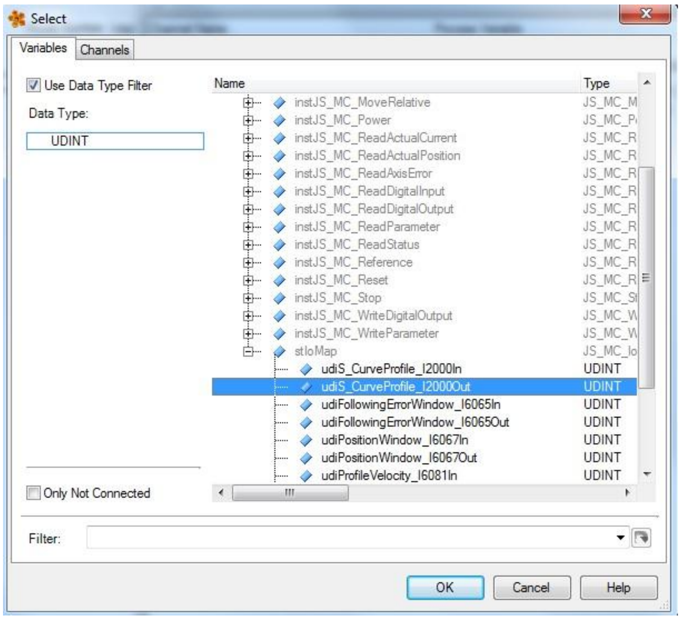
Channel Name	Process Variable	Data Type
ModuleOk		BOOL
S_CurveProfile_I2000Out	…	UDINT
ProcessStatusRegister_I2006		DINT
Controlword_I6040Out		UINT

The "Select" window will open.

You can open the according program which you like to use, with the symbol ⊞

In the tree layout of ⊟ „stIoMap" are all "Process Variables" which have a data type which matches the channel.

-> *OK*



Select

Variables | Channels

☑ Use Data Type Filter

Data Type: UDINT

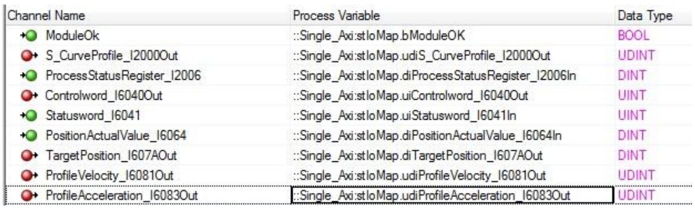
Name	Type
instJS_MC_MoveRelative	JS_MC_M
instJS_MC_Power	JS_MC_P
instJS_MC_ReadActualCurrent	JS_MC_R
instJS_MC_ReadActualPosition	JS_MC_R
instJS_MC_ReadAxisError	JS_MC_R
instJS_MC_ReadDigitalInput	JS_MC_R
instJS_MC_ReadDigitalOutput	JS_MC_R
instJS_MC_ReadParameter	JS_MC_R
instJS_MC_ReadStatus	JS_MC_R
instJS_MC_Reference	JS_MC_R
instJS_MC_Reset	JS_MC_R
instJS_MC_Stop	JS_MC_St
instJS_MC_WriteDigitalOutput	JS_MC_W
instJS_MC_WriteParameter	JS_MC_W
stIoMap	JS_MC_Io
udiS_CurveProfile_I2000In	UDINT
udiS_CurveProfile_I2000Out	UDINT
udiFollowingErrorWindow_I6065In	UDINT
udiFollowingErrorWindow_I6065Out	UDINT
udiPositionWindow_I6067In	UDINT
udiPositionWindow_I6067Out	UDINT
udiProfileVelocity_I6081In	UDINT

□ Only Not Connected

Filter:

After selecting the variable, the path of the variable can be seen in the I/O Mapping.

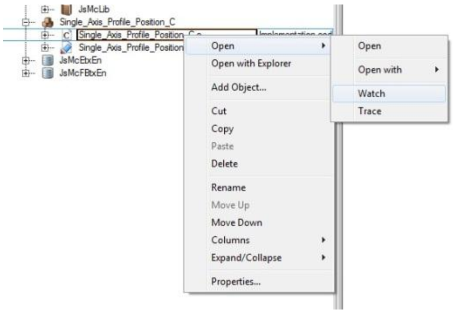
The assignment of a "Process Variable" has to be completed for every channel.



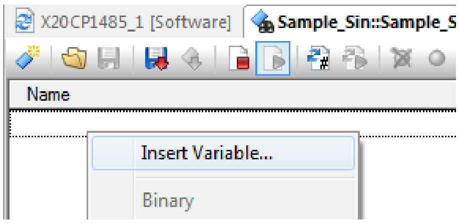
Channel Name	Process Variable	Data Type
ModuleOk	::Single_Axi:stIoMap.bModuleOK	BOOL
S_CurveProfile_I2000Out	::Single_Axi:stIoMap.udiS_CurveProfile_I2000Out	UDINT
ProcessStatusRegister_I2006	::Single_Axi:stIoMap.diProcessStatusRegister_I2006In	DINT
Controlword_I6040Out	::Single_Axi:stIoMap.uiControlword_I6040Out	UINT
Statusword_I6041	::Single_Axi:stIoMap.uiStatusword_I6041In	UINT
PositionActualValue_I6064	::Single_Axi:stIoMap.diPositionActualValue_I6064In	DINT
TargetPosition_I607AOut	::Single_Axi:stIoMap.diTargetPosition_I607AOut	DINT
ProfileVelocity_I6081Out	::Single_Axi:stIoMap.udiProfileVelocity_I6081Out	UDINT
ProfileAcceleration_I6083Out	::Single_Axi:stIoMap.udiProfileAcceleration_I6083Out	UDINT

5.6 Start Program Example

Logial View -> double click on the B&R PLC -> right click on the program example *„Single_Axis_Profile_Position_C“ -> Open -> Watch*



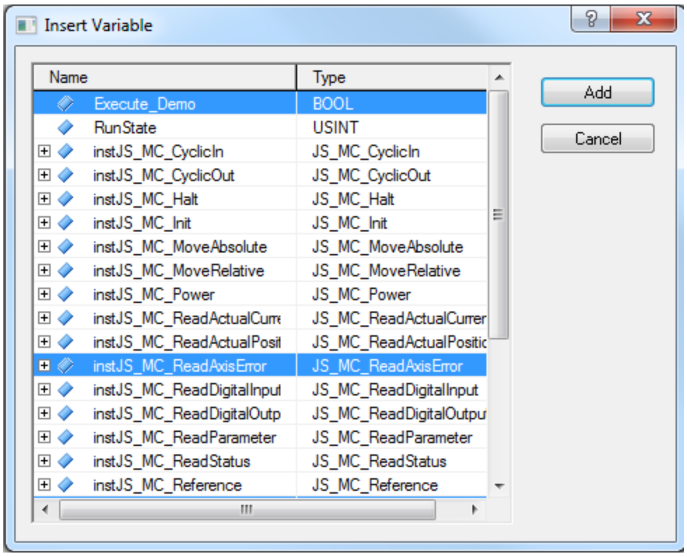
An empty window will be opened, right click -> *Insert Variable*



The window „Insert Variable“ will be opened. Every function block and variable which is used in the program example can be seen.

Select the required variables and function blocks. You will need at least the following:
Execute_Demo
instJS_MC_ReadAxisError
instJS_MC_Reset
szAxisErrorText

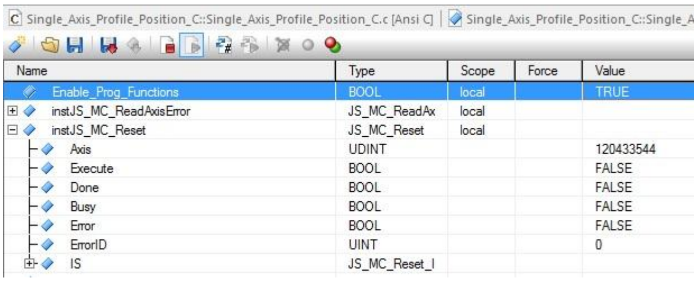
->Add



Name	Type
Execute_Demo	BOOL
RunState	USINT
instJS_MC_CyclicIn	JS_MC_CyclicIn
instJS_MC_CyclicOut	JS_MC_CyclicOut
instJS_MC_Halt	JS_MC_Halt
instJS_MC_Init	JS_MC_Init
instJS_MC_MoveAbsolute	JS_MC_MoveAbsolute
instJS_MC_MoveRelative	JS_MC_MoveRelative
instJS_MC_Power	JS_MC_Power
instJS_MC_ReadActualCurre	JS_MC_ReadActualCurrer
instJS_MC_ReadActualPosit	JS_MC_ReadActualPositic
instJS_MC_ReadAxisError	JS_MC_ReadAxisError
instJS_MC_ReadDigitalInput	JS_MC_ReadDigitalInput
instJS_MC_ReadDigitalOutp	JS_MC_ReadDigitalOutpu
instJS_MC_ReadParameter	JS_MC_ReadParameter
instJS_MC_ReadStatus	JS_MC_ReadStatus
instJS_MC_Reference	JS_MC_Reference

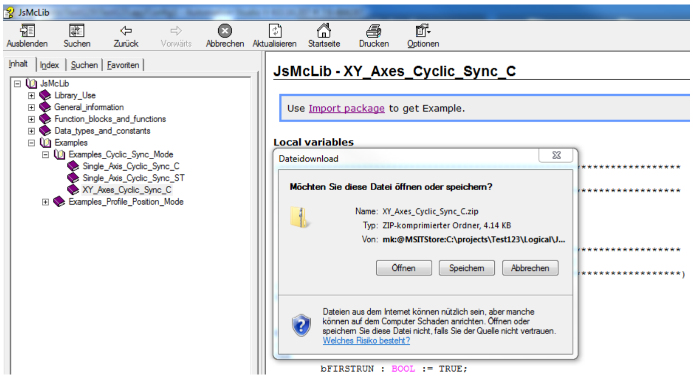
To Start the program example set the „Enable_Prog_Functions“ to TRUE. With FALSE the program is idling in a loop.

If an error occurs it can be quit by putting the variable “Execute” on TRUE in the function block “JS_MC_Reset”.


Single_Axis_Profile_Position_C::Single_Axis_Profile_Position_C.c [Ansi C]
Single_Axis_Profile_Position_C::Single_A

Name	Type	Scope	Force	Value
Enable_Prog_Functions	BOOL	local		TRUE
instJS_MC_ReadAxisError	JS_MC_ReadAx	local		
instJS_MC_Reset	JS_MC_Reset	local		
Axis	UDINT			120433544
Execute	BOOL			FALSE
Done	BOOL			FALSE
Busy	BOOL			FALSE
Error	BOOL			FALSE
ErrorID	UINT			0
IS	JS_MC_Reset_I			

6 Program example Cyclic Synchronous Position Mode



JsMcLib - XY_Axes_Cyclic_Sync_C

Use Import package to get Example.

Local variables

Möchten Sie diese Datei öffnen oder speichern?

Name: XY_Axes_Cyclic_Sync_C.zip
Typ: ZIP-komprimierter Ordner, 4.14 KB
Von: mk:@MSITStore:C:\projects\Test123\Logical\J...

Dateien aus dem Internet können nützlich sein, aber manche können auf dem Computer Schaden anrichten. Öffnen oder speichern Sie diese Datei nicht, falls Sie der Quelle nicht vertrauen. Welches Risiko besteht?

```
bFIRSTRUN : BOOL := TRUE;
```

The program example is stored as c-coder or structured text in the user's help of the JsMcLib. Firstly the example needs to be saved to the PC and then it can be imported to the project.

Logical View, select JsMcLib Library, press F1 to open the user's help.

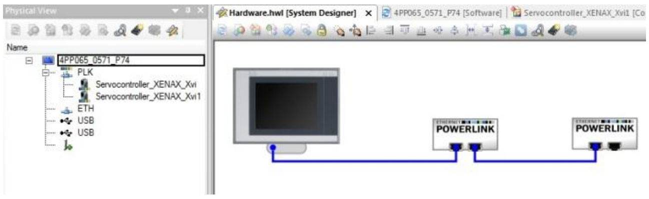
Under "Examples" the examples program can be found.

In our Example we choose "XY_Axes_Cyclic_Sync_C" and click on "Import Package".

Then we save this file on the PC and extract the .zip-file. Right click -> "extract all"

This program example works with an X-Y cross table and interpolates a circle. Therefore two XENAX® Servo Controllers are needed.

6.1 Integrate second XENAX® Servo Controller in project.



Physical View
Name
4PP065_0571_P74
PLK
Servocontroller_XENAX_Xvi
Servocontroller_XENAX_Xvi1
ETH
USB
USB
Hardware.hwl [System Designer]
4PP065_0571_P74 [Software]
Servocontroller_XENAX_Xvi1 [Co
POWERLINK
POWERLINK

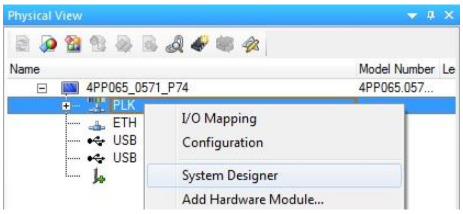
Integrate an additional XENAX® Servo Controller in project like seen above under „ Embed XEANAX® Servo Controller in project"

Please note that the node numbers (1 and 2) have to be different in the program code.

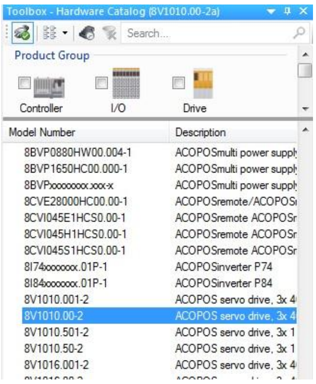
6.2 Include virtual Axis

For the **Cyclic Synchronous Position Mode**, a virtual axis is needed in the Automation Studio. The Virtual Axis calculates and transmits a new position for the XENAX® Servo Controller every [ms]

In *Physical View* select right click the power link interface (PLK) -> *System Designer.*

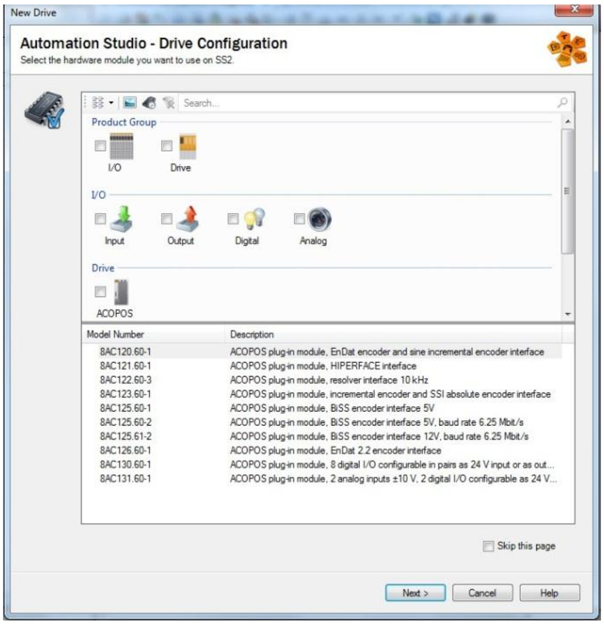


From the “Toolbox Hardware Catalog” add the „AOPOS servo drive 8V1010.00-2“ by double click.



Model Number	Description
8BVP0880HW00.004-1	ACOPOSmulti power suppl
8BVP1650HC00.000-1	ACOPOSmulti power suppl
8BVPxxxxxxxx.xxx-x	ACOPOSmulti power suppl
8CVE28000HC00.00-1	ACOPOSremote/ACOPOSı
8CVI045E1HCS0.00-1	ACOPOSremote ACOPOSr
8CVI045H1HCS0.00-1	ACOPOSremote ACOPOSr
8CVI045S1HCS0.00-1	ACOPOSremote ACOPOSr
8I74xxxxxxx.01P-1	ACOPOSinverter P74
8I84xxxxxxx.01P-1	ACOPOSinverter P84
8V1010.001-2	ACOPOS servo drive, 3x 4
8V1010.00-2	ACOPOS servo drive, 3x 4
8V1010.501-2	ACOPOS servo drive, 3x 1
8V1010.50-2	ACOPOS servo drive, 3x 1
8V1016.001-2	ACOPOS servo drive, 3x 4

The window „New Drive“ can be confirmed three times with *->Next.*



Automation Studio - Drive Configuration

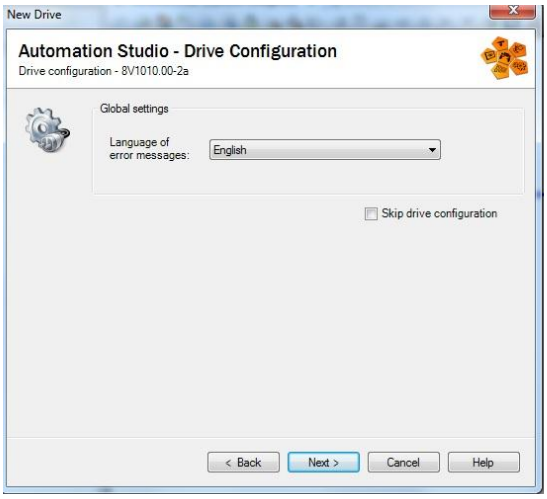
Select the hardware module you want to use on SS2.

Model Number	Description
8AC120.60-1	ACOPOS plug-in module, EnDat encoder and sine incremental encoder interface
8AC121.60-1	ACOPOS plug-in module, HIPERFACE interface
8AC122.60-3	ACOPOS plug-in module, resolver interface 10 kHz
8AC123.60-1	ACOPOS plug-in module, incremental encoder and SSI absolute encoder interface
8AC125.60-1	ACOPOS plug-in module, BiSS encoder interface 5V
8AC125.60-2	ACOPOS plug-in module, BiSS encoder interface 5V, baud rate 6.25 Mbit/s
8AC125.61-2	ACOPOS plug-in module, BiSS encoder interface 12V, baud rate 6.25 Mbit/s
8AC126.60-1	ACOPOS plug-in module, EnDat 2.2 encoder interface
8AC130.60-1	ACOPOS plug-in module, 8 digital I/O configurable in pairs as 24 V input or as out...
8AC131.60-1	ACOPOS plug-in module, 2 analog inputs ±10 V, 2 digital I/O configurable as 24 V...

□ Skip this page

Choose the language for the error messages of the virtual axis.

->Next



New Drive

Automation Studio - Drive Configuration

Drive configuration - 8V1010.00-2a

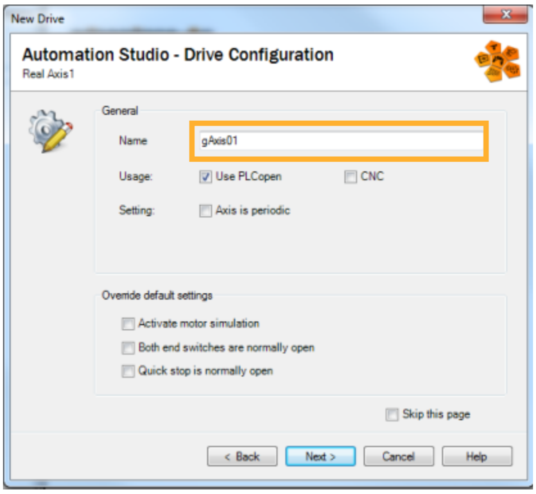
Global settings

Language of error messages: English

□ Skip drive configuration

Write a name for the virtual axis.

->Next



New Drive

Automation Studio - Drive Configuration

Real Axis1

General

Name: gAxis01

Usage: ☑ Use PLCopen □ CNC

Setting: □ Axis is periodic

Override default settings

□ Activate motor simulation

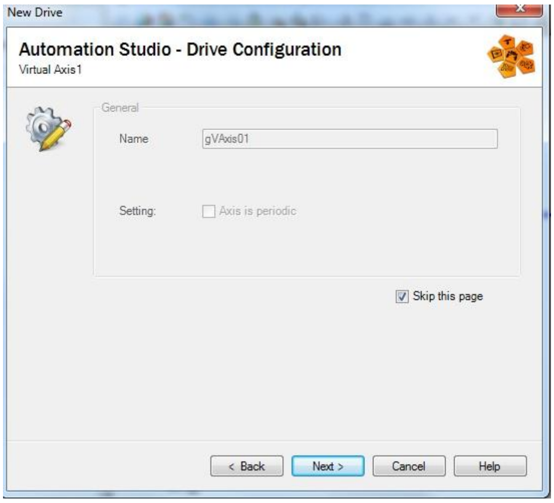
□ Both end switches are normally open

□ Quick stop is normally open

□ Skip this page

Confirm this window with

->Next



New Drive

Automation Studio - Drive Configuration

Virtual Axis1

General

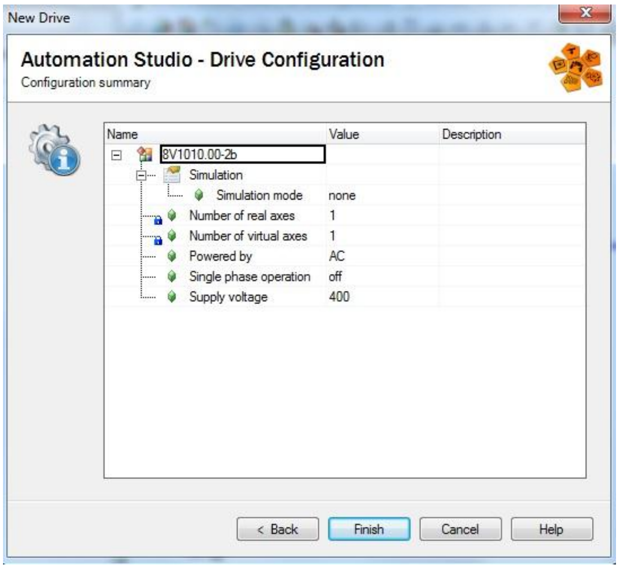
Name: gVAxis01

Setting: □ Axis is periodic

☑ Skip this page

Confirm the „Configuration summary“ with

->Finish



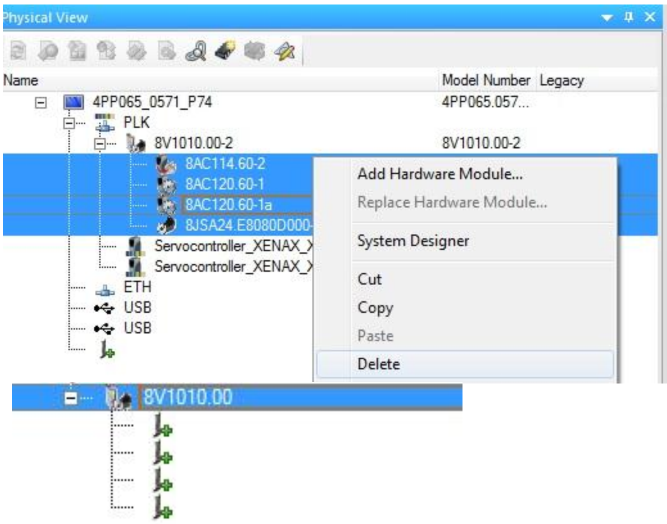
New Drive

Automation Studio - Drive Configuration

Configuration summary

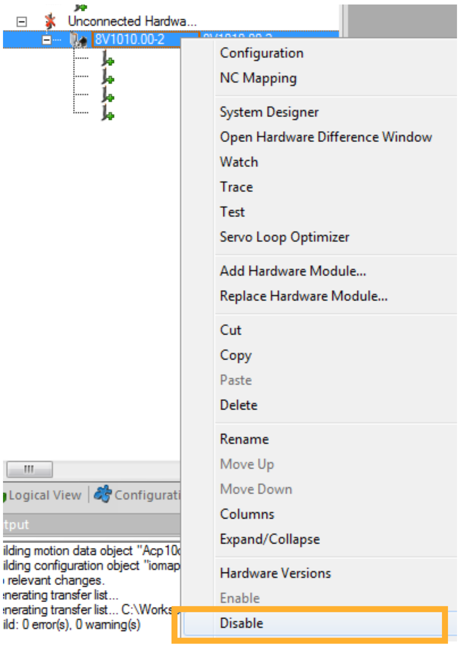
Name	Value	Description
8V1010.00-2b		
Simulation		
Simulation mode	none	
Number of real axes	1	
Number of virtual axes	1	
Powered by	AC	
Single phase operation	off	
Supply voltage	400	

In the *Physical View* under the B&R PLC (here *X20CP1485-1*) the new virtual Axis with all its functionalities can be seen.
Because we only use this Axis virtual, we need to delete all the function objects.
Select all -> right click-> Delete



Name	Model Number	Legacy
4PP065_0571_P74	4PP065.057...	
PLK		
8V1010.00-2	8V1010.00-2	
8AC114.60-2		
8AC120.60-1		
8AC120.60-1a		
8JSA24.E8080D000-		
Servocontroller_XENAX_X		
Servocontroller_XENAX_X		
ETH		
USB		
USB		

Because there is no real connection between the B&R PLC and the virtual Axis, we need to disable the Axis.
Right click -> Disable

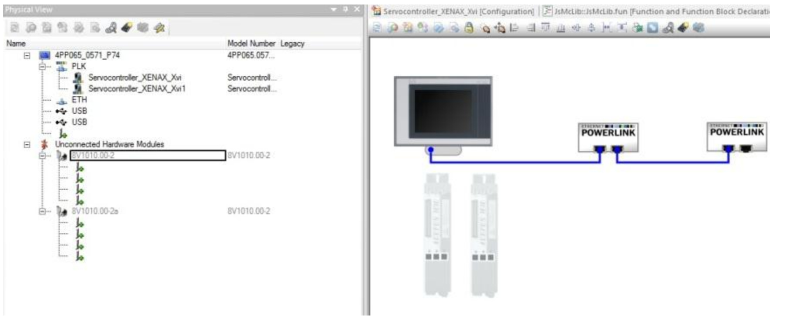


Unconnected Hardwa...

8V1010.00-2

```
ilding motion data object "Acp10
ilding configuration object "iomap
 relevant changes.
nerating transfer list...
nerating transfer list... C:\Works
ild: 0 error(s), 0 warning(s)
```

Note:
This example works with 2 Axis.
A second virtual Axis has to be included.

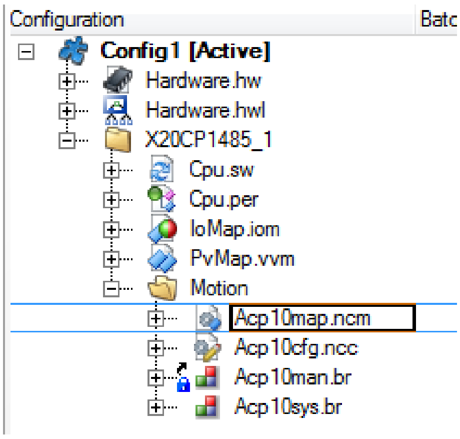


```
Physical View
Name                                      Model Number  Legacy
⊟ 4PP065_0571_P74                         4PP065.057...
  ⊟ PLK
      Servocontroller_XENAX_Xvi           Servocontroll...
      Servocontroller_XENAX_Xvi1          Servocontroll...
    ETH
    USB
    USB
⊟ Unconnected Hardware Modules
  ⊟ 8V1010.00-2                           8V1010.00-2
  ⊟ 8V1010.00-2a                          8V1010.00-2
```

Servocontroller_XENAX_Xvi [Configuration] | JsMcLib::JsMcLib.fun [Function and Function Block Declarati

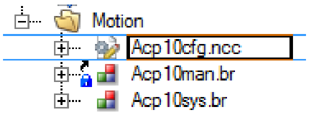
6.3 Insert interface for virtual axis

In the *Configurations View* under *X20CP1485_1 -> Motion.* Mark “Acp10map.ncm“ and delete.



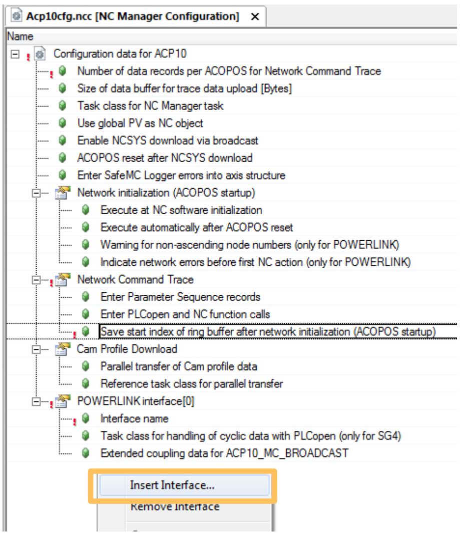
```
Configuration                 Batc
⊟ Config1 [Active]
  ⊞ Hardware.hw
  ⊞ Hardware.hwl
  ⊟ X20CP1485_1
    ⊞ Cpu.sw
    ⊞ Cpu.per
    ⊞ IoMap.iom
    ⊞ PvMap.vvm
    ⊟ Motion
      ⊞ Acp10map.ncm
      ⊞ Acp10cfg.ncc
      ⊞ Acp10man.br
      ⊞ Acp10sys.br
```

To open the NC Manager configuration double click on “Acp10cfg.ncc”.



```
⊟ Motion
  ⊞ Acp10cfg.ncc
  ⊞ Acp10man.br
  ⊞ Acp10sys.br
```

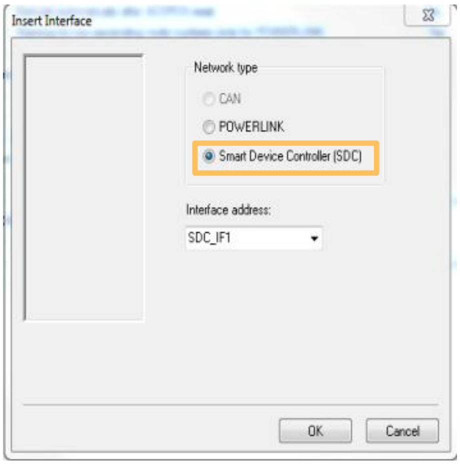
Right click on an empty spot in the NC manager Configuration -> Insert Interface…



```
Acp10cfg.ncc [NC Manager Configuration] ×
Name
⊟ Configuration data for ACP10
    Number of data records per ACOPOS for Network Command Trace
    Size of data buffer for trace data upload [Bytes]
    Task class for NC Manager task
    Use global PV as NC object
    Enable NCSYS download via broadcast
    ACOPOS reset after NCSYS download
    Enter SafeMC Logger errors into axis structure
  ⊟ Network initialization (ACOPOS startup)
      Execute at NC software initialization
      Execute automatically after ACOPOS reset
      Warning for non-ascending node numbers (only for POWERLINK)
      Indicate network errors before first NC action (only for POWERLINK)
  ⊟ Network Command Trace
      Enter Parameter Sequence records
      Enter PLCopen and NC function calls
      Save start index of ring buffer after network initialization (ACOPOS startup)
  ⊟ Cam Profile Download
      Parallel transfer of Cam profile data
      Reference task class for parallel transfer
  ⊟ POWERLINK interface[0]
      Interface name
      Task class for handling of cyclic data with PLCopen (only for SG4)
      Extended coupling data for ACP10_MC_BROADCAST
```

Select Smart Device Controller (SDC) and confirm with *OK*.

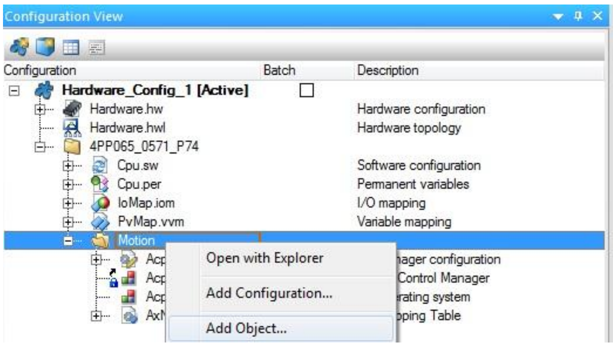
Note: This step has to be made twice, once for each axis.



6.3.1 Insert the NC-Mapping Table

The NC Mapping table will be used for every virtual axis and needs to be inserted only once.

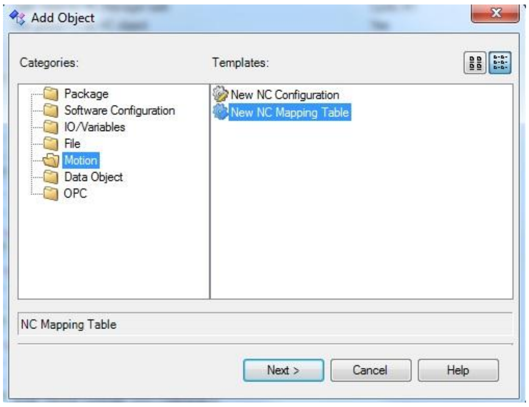
Under *Configuration View -> X20CP1485_1 -> right click on Motion -> Add Object...*


Configuration View

Configuration	Batch	Description
Hardware_Config_1 [Active]	□	
Hardware.hw		Hardware configuration
Hardware.hwl		Hardware topology
4PP065_0571_P74		
Cpu.sw		Software configuration
Cpu.per		Permanent variables
IoMap.iom		I/O mapping
PvMap.vvm		Variable mapping
Motion		
Acp		ager configuration
Acp		Control Manager
Acp		rating system
AxI		pping Table

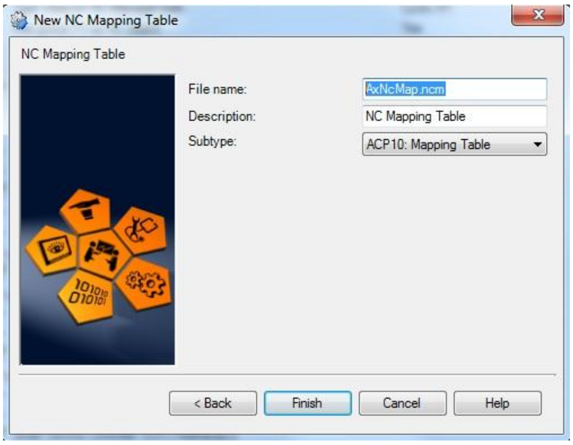
Categories -> Motion -> Templates -> New NC Mapping Table

-> Next



Insert name *„AxNcMap.ncm“*

-> Finish

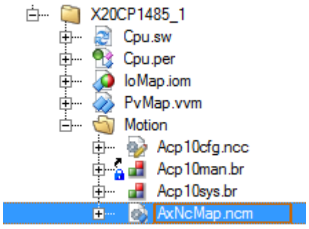


File name: AxNcMap.ncm

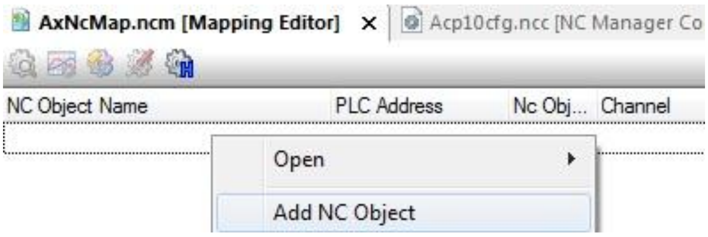
Description: NC Mapping Table

Subtype: ACP10: Mapping Table

To open The NC Mapping Table double click *AxNcMap.ncm* in the “*Configuration View*”.

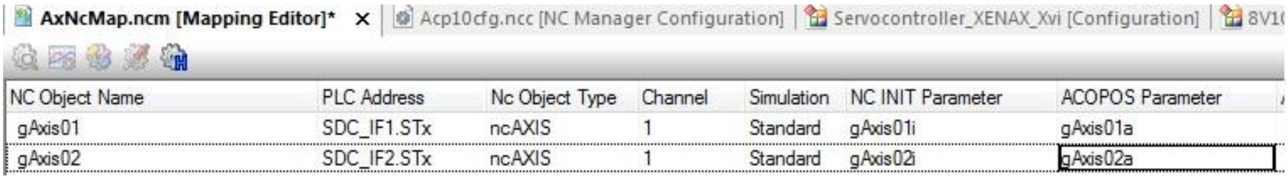


Right click “*Add NC Object*”



The following parameters need to be set for the NC object.

„NC Object Name“ is similar to the name of the inserted virtual Axis.
This step is also taken twice


AxNcMap.ncm [Mapping Editor]*
Acp10cfg.ncc [NC Manager Configuration]
Servocontroller_XENAX_Xvi [Configuration]
8V1

NC Object Name	PLC Address	Nc Object Type	Channel	Simulation	NC INIT Parameter	ACOPOS Parameter
gAxis01	SDC_IF1.STx	ncAXIS	1	Standard	gAxis01i	gAxis01a
gAxis02	SDC_IF2.STx	ncAXIS	1	Standard	gAxis02i	gAxis02a

6.3.2 Parameter-Adjustments

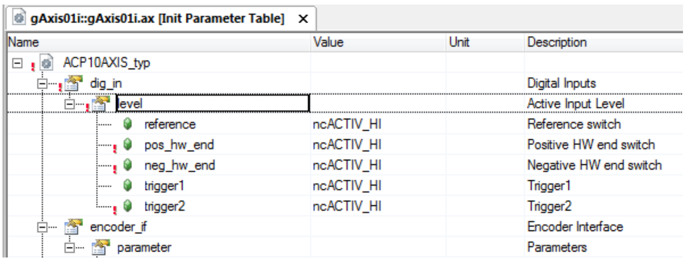
Open the *Logical View* select the inserted virtual Axis:
gAxisXXobj -> gaxisXXi double click.

The following parameter need to be adjusted:

Input function

ACP10AXIS_typ -> dig_in -> level

Adjust the Input functions as follows:
pos_hw_end to „ncACTIV_HI"
neg_hw_end to „ncACTIV_HI"
trigger2 to „ncACTIV_HI"



gAxis01i::gAxis01i.ax [Init Parameter Table]

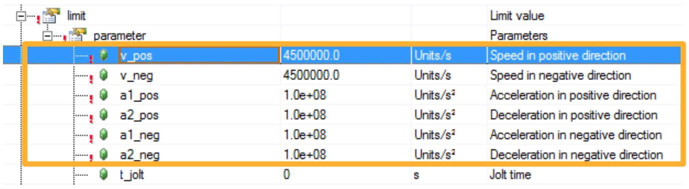
Name	Value	Unit	Description
ACP10AXIS_typ			
dig_in			Digital Inputs
level			Active Input Level
reference	ncACTIV_HI		Reference switch
pos_hw_end	ncACTIV_HI		Positive HW end switch
neg_hw_end	ncACTIV_HI		Negative HW end switch
trigger1	ncACTIV_HI		Trigger1
trigger2	ncACTIV_HI		Trigger2
encoder_if			Encoder Interface
parameter			Parameters

Velocity, Acceleration and brake ramp

Set the maximum velocity (Speed), maximum acceleration (Acceleration) and maximum brake ramp (Deceleration), for the Project.

ACP10AXIS_typ -> limit -> parameter

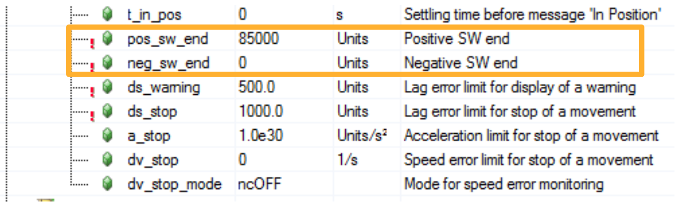
Speed (**v_pos** und **v_neg**),4'500'000
Acceleration (**a1_pos** und **a1_neg**),1.0e + 8
Deceleration (**a2_pos** und **a2_neg**),1.0e + 8
Set to the maximum of the linear motor axis.



limit			Limit value
parameter			Parameters
v_pos	4500000.0	Units/s	Speed in positive direction
v_neg	4500000.0	Units/s	Speed in negative direction
a1_pos	1.0e+08	Units/s^2	Acceleration in positive direction
a2_pos	1.0e+08	Units/s^2	Deceleration in positive direction
a1_neg	1.0e+08	Units/s^2	Acceleration in negative direction
a2_neg	1.0e+08	Units/s^2	Deceleration in negative direction
t_jolt	0	s	Jolt time

Stroke

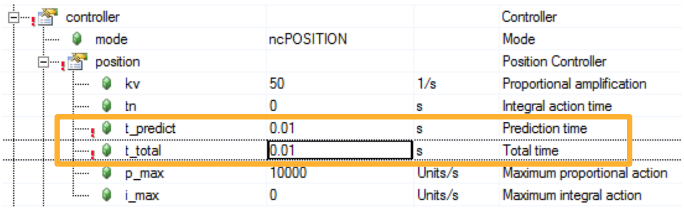
The Stroke has to match the mechanical limits of the linear motor.
In our example a ELAX® Ex30F20 is used. The maximum stroke of it is 30000 µm:
neg_sw_end = 0
pos_sw_end = 30000



t_in_pos	0	s	Settling time before message 'In Position'
pos_sw_end	85000	Units	Positive SW end
neg_sw_end	0	Units	Negative SW end
ds_warning	500.0	Units	Lag error limit for display of a warning
ds_stop	1000.0	Units	Lag error limit for stop of a movement
a_stop	1.0e30	Units/s^2	Acceleration limit for stop of a movement
dv_stop	0	1/s	Speed error limit for stop of a movement
dv_stop_mode	ncOFF		Mode for speed error monitoring

Time

Time adjustment
ACP10AXIS_typ -> controller-> position
t_predict
t_total



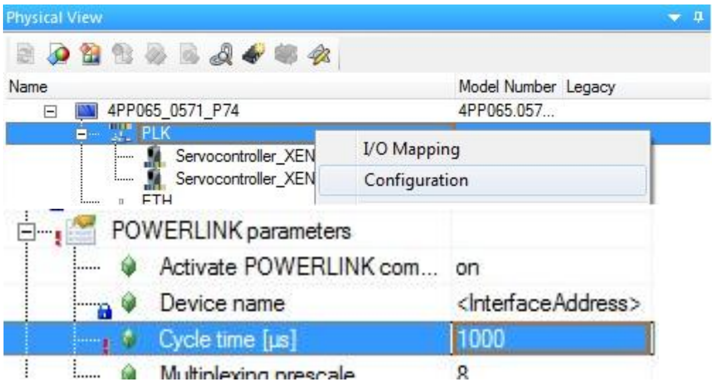
controller			Controller
mode	ncPOSITION		Mode
position			Position Controller
kv	50	1/s	Proportional amplification
tn	0	s	Integral action time
t_predict	0.01	s	Prediction time
t_total	0.01	s	Total time
p_max	10000	Units/s	Maximum proportional action
i_max	0	Units/s	Maximum integral action

Note: This parameter adjustment has to be done for every virtual Axis.

6.3.3 Cycle time

The time used should be as short as possible.
Physical View -> PLK -> right click -> Configuration

The cycle time is set to 1000[µs]



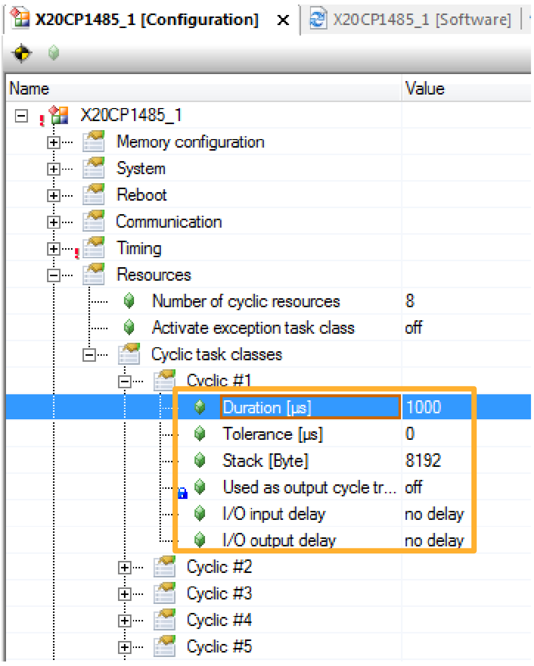
POWERLINK parameters	
Activate POWERLINK com...	on
Device name	<InterfaceAddress>
Cycle time [µs]	1000
Multiplexing prescale	8

Physical View -> double click on the B&R PLC(here *X20CP1485-1*) -> right click *Cyclic #1* -> *Properties*.

Set the duration for the virtual Axis in the Cyclic#1 to 1000 µs.

The Duration must be the same as the Powerlink Cycle Time which was set before.
The "Tolerance" must be 0ms and "*I/O outputs delay*" has to be set to „*no delay*

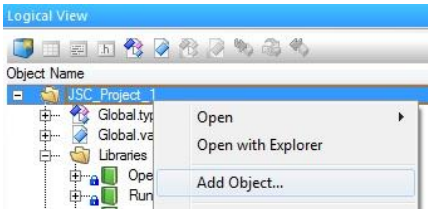
Note: If these settings aren't made like explained, the initialisation will be stopped by an error and a commissioning of the virtual axis won't be possible.



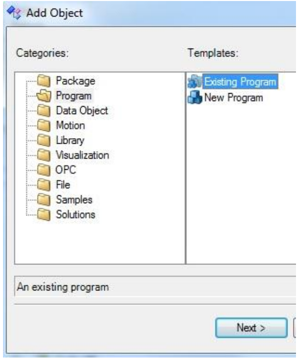
Name	Value
X20CP1485_1	
Memory configuration	
System	
Reboot	
Communication	
Timing	
Resources	
Number of cyclic resources	8
Activate exception task class	off
Cyclic task classes	
Cyclic #1	
Duration [µs]	1000
Tolerance [µs]	0
Stack [Byte]	8192
Used as output cycle tr...	off
I/O input delay	no delay
I/O output delay	no delay
Cyclic #2	
Cyclic #3	
Cyclic #4	
Cyclic #5	

6.4 Insert Program Example

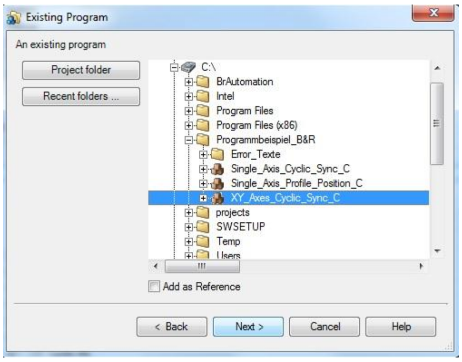
Logical View -> JSC_Project_1 -> right click-> Add Object…



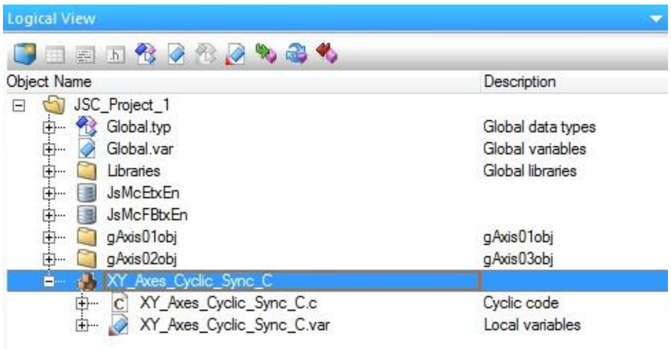
Categories: program ->Existing program -> Next



Select the path of the *"XY_Axes_Cyclic_Sync_C".*
-> Next



The program can be seen now in the Logic*al View*.

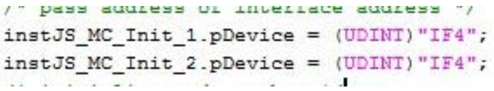


Object Name	Description
JSC_Project_1	
Global.typ	Global data types
Global.var	Global variables
Libraries	Global libraries
JsMcEtxEn	
JsMcFBtxEn	
gAxis01obj	gAxis01obj
gAxis02obj	gAxis03obj
XY_Axes_Cyclic_Sync_C	
XY_Axes_Cyclic_Sync_C.c	Cyclic code
XY_Axes_Cyclic_Sync_C.var	Local variables

6.5 Configure Ethernet Interface

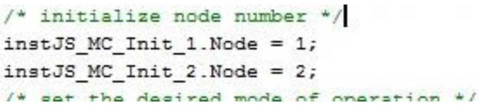
With double click on „XY_Axes_Cyclic_Sync_C.c", the program code will be opened.
In the code are the Ethernet Powerlink Interface Adress (Ifx) the node number and the used error object.

Ethernet Powerlink Address EPL:
Adapt the line
`instJS_MC_Init.pDevice = (UDINT)”IF4”;`.
The Powerlink address is for both XENAX® Servo Controller the same.



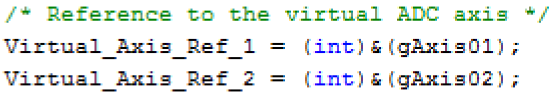
```
/* pass address of interface address */
instJS_MC_Init_1.pDevice = (UDINT)"IF4";
instJS_MC_Init_2.pDevice = (UDINT)"IF4";
```

Node Nr.
The node number is defined in the line
`instJS_MC_Init.Node= 1` (here 1 & 2).
The node number has to be the same as the CI (Card Identifier) of the XENAX® Servo Controller.
The CI is explained above.

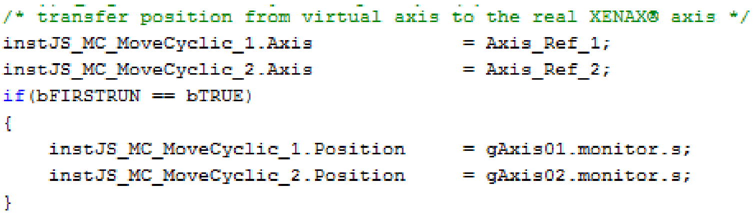


```
/* initialize node number */
instJS_MC_Init_1.Node = 1;
instJS_MC_Init_2.Node = 2;
/* set the desired mode of operation */
```

Coupling the virtual Axis
In the row:
`Virtual_Axis_Ref_1 = (int) & (gAxis01);`
und `instJS_MC_MoveCyclic_1.Axis = Axis_Ref_1;`
the axis names (gAxis01 & gAxis02) are visible.
The name in the program code hase to correspond the axis name to succesfully couple the virtual axis.



```
/* Reference to the virtual ADC axis */
Virtual_Axis_Ref_1 = (int)&(gAxis01);
Virtual_Axis_Ref_2 = (int)&(gAxis02);
```

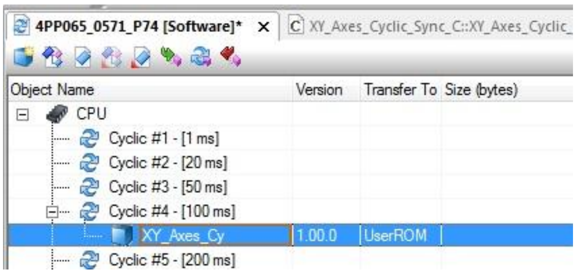


```
/* transfer position from virtual axis to the real XENAX® axis */
instJS_MC_MoveCyclic_1.Axis             = Axis_Ref_1;
instJS_MC_MoveCyclic_2.Axis             = Axis_Ref_2;
if(bFIRSTRUN == bTRUE)
{
    instJS_MC_MoveCyclic_1.Position     = gAxis01.monitor.s;
    instJS_MC_MoveCyclic_2.Position     = gAxis02.monitor.s;
}
```

6.6 Move Program Example to Task with Desired Cycle Time

Under Physical *View -> B&R PLC -> double click ->* Software Configuration

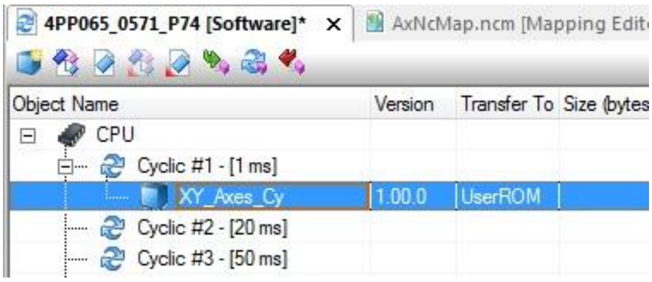
The program example is always in Cyclic #4(100ms) as default.



4PP065_0571_P74 [Software]* ×
XY_Axes_Cyclic_Sync_C::XY_Axes_Cyclic_

Object Name	Version	Transfer To	Size (bytes)
CPU			
Cyclic #1 - [1 ms]			
Cyclic #2 - [20 ms]			
Cyclic #3 - [50 ms]			
Cyclic #4 - [100 ms]			
XY_Axes_Cy	1.00.0	UserROM	
Cyclic #5 - [200 ms]			

We move the program example to the „Cyclic #1“, which has a shorter cycle time (1ms).

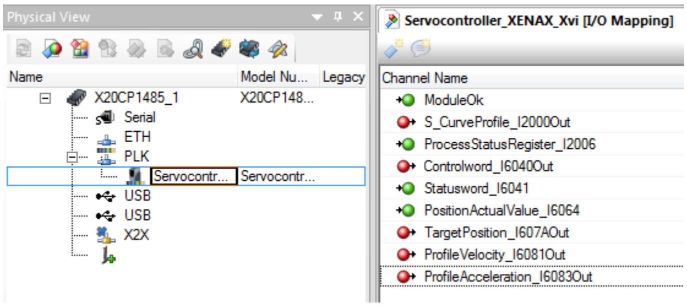


Object Name	Version	Transfer To	Size (bytes
CPU			
Cyclic #1 - [1 ms]			
XY_Axes_Cy	1.00.0	UserROM	
Cyclic #2 - [20 ms]			
Cyclic #3 - [50 ms]			

6.7 I/O Mapping, connect the .xdd Channels with the program variables.

To use the .xdd-Channels (cyclic data of the XENAX®) in the library they have to be connected to a “Process Variable”.

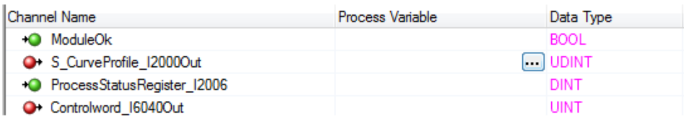
Physical View, double click on XENAX® Servo Controller -> *I/O Mapping*.



Name	Model Nu...	Legacy
X20CP1485_1	X20CP148...	
Serial		
ETH		
PLK		
Servocontr...	Servocontr...	
USB		
USB		
X2X		

Channel Name
ModuleOk
S_CurveProfile_I2000Out
ProcessStatusRegister_I2006
Controlword_I6040Out
Statusword_I6041
PositionActualValue_I6064
TargetPosition_I607AOut
ProfileVelocity_I6081Out
ProfileAcceleration_I6083Out

Select in the row „Process Variable“ the cell of the desired Channel (e.g.: S_CurveProfile_I2000Out). Click on the … Symbol.



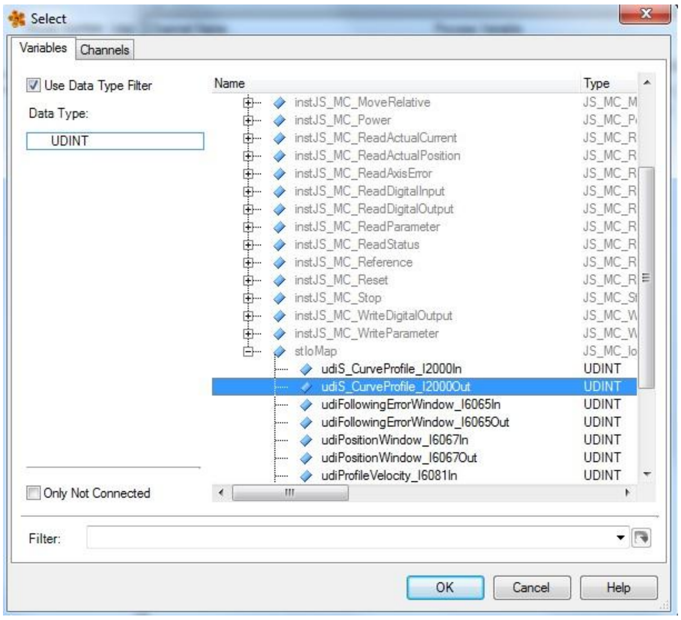
Channel Name	Process Variable	Data Type
ModuleOk		BOOL
S_CurveProfile_I2000Out	…	UDINT
ProcessStatusRegister_I2006		DINT
Controlword_I6040Out		UINT

The Window “Select” will be opened

Please open the according program with the ⊞ symbol.

In the tree layout of ⊟ „stIoMap“ there are all “Process Variables” which have a data type that matches the channel. Select the corresponding variables for the channel.

-> *OK*



☑ Use Data Type Filter

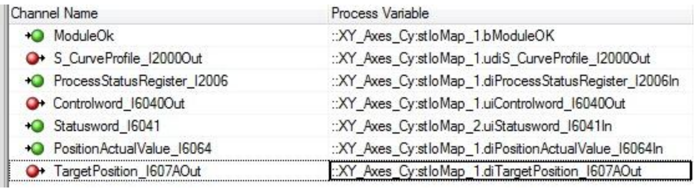
Data Type: UDINT

☐ Only Not Connected

Filter:

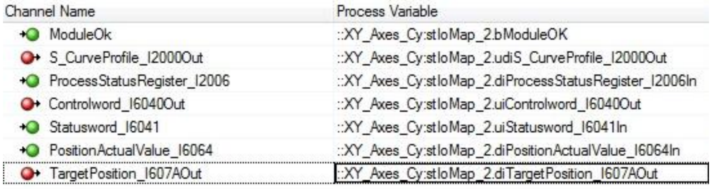
Name	Type
instJS_MC_MoveRelative	JS_MC_M
instJS_MC_Power	JS_MC_P
instJS_MC_ReadActualCurrent	JS_MC_R
instJS_MC_ReadActualPosition	JS_MC_R
instJS_MC_ReadAxisError	JS_MC_R
instJS_MC_ReadDigitalInput	JS_MC_R
instJS_MC_ReadDigitalOutput	JS_MC_R
instJS_MC_ReadParameter	JS_MC_R
instJS_MC_ReadStatus	JS_MC_R
instJS_MC_Reference	JS_MC_R
instJS_MC_Reset	JS_MC_R
instJS_MC_Stop	JS_MC_St
instJS_MC_WriteDigitalOutput	JS_MC_W
instJS_MC_WriteParameter	JS_MC_W
stIoMap	JS_MC_Io
udiS_CurveProfile_I2000In	UDINT
udiS_CurveProfile_I2000Out	UDINT
udiFollowingErrorWindow_I6065In	UDINT
udiFollowingErrorWindow_I6065Out	UDINT
udiPositionWindow_I6067In	UDINT
udiPositionWindow_I6067Out	UDINT
udiProfileVelocity_I6081In	UDINT

After the process variables are selected, the path can be seen in the “I/O Mapping”.



Channel Name	Process Variable
ModuleOk	::XY_Axes_Cy:stIoMap_1.bModuleOK
S_CurveProfile_I2000Out	::XY_Axes_Cy:stIoMap_1.udiS_CurveProfile_I2000Out
ProcessStatusRegister_I2006	::XY_Axes_Cy:stIoMap_1.diProcessStatusRegister_I2006In
Controlword_I6040Out	::XY_Axes_Cy:stIoMap_1.uiControlword_I6040Out
Statusword_I6041	::XY_Axes_Cy:stIoMap_2.uiStatusword_I6041In
PositionActualValue_I6064	::XY_Axes_Cy:stIoMap_1.diPositionActualValue_I6064In
TargetPosition_I607AOut	::XY_Axes_Cy:stIoMap_1.diTargetPosition_I607AOut

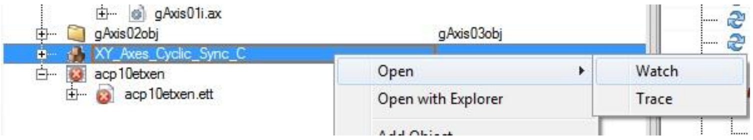
Every channel needs to be connected to a process variable. This has to be done for every XENAX®



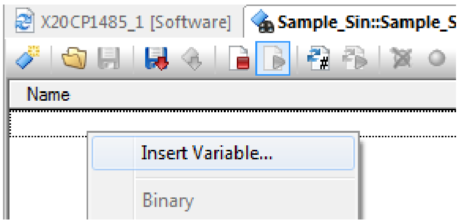
Channel Name	Process Variable
ModuleOk	::XY_Axes_Cy:stIoMap_2.bModuleOK
S_CurveProfile_I2000Out	::XY_Axes_Cy:stIoMap_2.udiS_CurveProfile_I2000Out
ProcessStatusRegister_I2006	::XY_Axes_Cy:stIoMap_2.diProcessStatusRegister_I2006In
Controlword_I6040Out	::XY_Axes_Cy:stIoMap_2.uiControlword_I6040Out
Statusword_I6041	::XY_Axes_Cy:stIoMap_2.uiStatusword_I6041In
PositionActualValue_I6064	::XY_Axes_Cy:stIoMap_2.diPositionActualValue_I6064In
TargetPosition_I607AOut	::XY_Axes_Cy:stIoMap_2.diTargetPosition_I607AOut

6.8 Start Program Example

Logial View -> double click on the B&R PLC (here X20CP1485-1) -> right click on the program example *„XY_Axes_Cyclic_Sync_C“-> Open -> Watch*



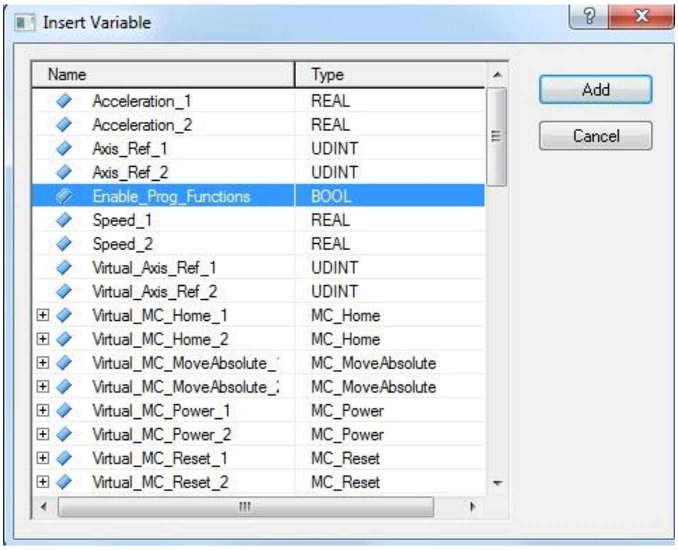
An empty window will be opened, right click -> *Insert Variable…*



The *„Insert Variable“* window is opened with all the function blocks and variables of the program example.

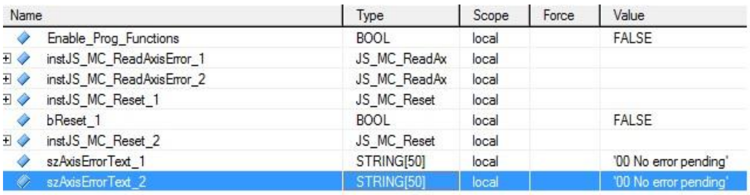
Select the required variables and function blocks. You will need at least the following:
Enable_Prog_Functions
instJS_MC_ReadAxisError_1
instJS_MC_ReadAxisError_2
instJS_MC_Reset_1
instJS_MC_Reset_2
szAxisErrorText_1
szAxisErrorText_2
bReset_1

->Add



Name	Type
Acceleration_1	REAL
Acceleration_2	REAL
Axis_Ref_1	UDINT
Axis_Ref_2	UDINT
Enable_Prog_Functions	BOOL
Speed_1	REAL
Speed_2	REAL
Virtual_Axis_Ref_1	UDINT
Virtual_Axis_Ref_2	UDINT
Virtual_MC_Home_1	MC_Home
Virtual_MC_Home_2	MC_Home
Virtual_MC_MoveAbsolute_	MC_MoveAbsolute
Virtual_MC_MoveAbsolute_:	MC_MoveAbsolute
Virtual_MC_Power_1	MC_Power
Virtual_MC_Power_2	MC_Power
Virtual_MC_Reset_1	MC_Reset
Virtual_MC_Reset_2	MC_Reset

Set the “Enable_Prog_Functions” to TRUE to start the program. With FALSE the program is idling in a loop.



Name	Type	Scope	Force	Value
Enable_Prog_Functions	BOOL	local		FALSE
instJS_MC_ReadAxisError_1	JS_MC_ReadAx	local		
instJS_MC_ReadAxisError_2	JS_MC_ReadAx	local		
instJS_MC_Reset_1	JS_MC_Reset	local		
bReset_1	BOOL	local		FALSE
instJS_MC_Reset_2	JS_MC_Reset	local		
szAxisErrorText_1	STRING[50]	local		'00 No error pending'
szAxisErrorText_2	STRING[50]	local		'00 No error pending'

A Error can be quit by setting the bReset_1 to TRUE.

Note:
In the help of the library there is a program example with 1 Axis in Cyclic Synchronized mode.

Notes

This instruction manual contains copyright protected information. All rights are reserved. This document may not be in its entirety or partially copied, duplicated or translated without the prior consent of Jenny Science AG.

Jenny Science AG grants no guarantee on, or will be held responsible for, any incidents resulting from false information.

Information in this instruction manual is subject to change.

Jenny Science AG
Sandblatte 7a
CH-6026 Rain, Schweiz

Tel +41 (0) 41 455 44 55
Fax +41 (0) 41 455 44 50

www.jennyscience.ch
info@jennyscience.ch